

EleOffice user manual

Covering these products:

EleRobot

Automatic data manipulation and calculation tool

EleExxel

Product and price list generator for publishing Elestore database content to nicely designed MSeExcel sheets formatted from used-defined templates

ElePricelist

Product and price list generator for output to MSWord,y MSeExcel and/or HTML the simple and very effective database publishing system for all MS Office users

EleProductSite

HTML generator creates entire websites with individually designed product pages, complete navigation and bi-directional links to and from EleAIS server content

All based on the

Elestore

Product Content Management System
for multiple re-use of the Elestore database content

Handbook and User's Manual

This manual is still work in progress. It was created using voice recognition software and was not yet proof-read.
This is where some odd mistakes may result from. We apologize for some inconveniences!

Last revision on the 20.02.2009 (#095)

© Heinz-Geerd Neumann

Table of content

EleOffice user manual.....	1
TABLE OF CONTENT	2
OVERVIEW ON ELEOFFICE AND ELEFAMILY	5
EleFamily	5
Stand-alone publishing products and extensions to Elestore.....	5
ELEROBOT.....	8
Overview	8
EleRobot brief abstract	8
Versions of EleRobot	8
EleRobot projects.....	8
Overview on EleRobot projects	8
EleRobot project table	9
The EleRobot project details form	10
The settings of an EleRobot project	11
Running EleRobot.....	12
Attribute naming conventions	13
EleRobot templates – structure, format, data references.....	14
What are EleRobot templates?	14
EleRobot and user-defined templates	14
The structure of an EleRobot template file.....	14
The types of references and data in cells in EleRobot templates.....	15
EleRobot template samples explained.....	17
Explanations in EleRobot template files	17
A EleRobot template sample	17
Another EleRobot template sample	19
EleRobot data change protocol file	21
Overview and purpose.....	21
Sample of an EleRobot data change protocol file	21
ELEEXCEL.....	23
Overview on EleExxel.....	23
EleExxel brief abstract	23
How EleExxel is used.....	23
Versions of EleExxel.....	24
EleExxel’s three types of templates - overview	25
User-defined templates in EleExxel	26
EleExxel templates – structure and data references.....	27
What are EleExxel templates?	27
The types of references and data in EleExxel template cells.....	27
Row type identifiers in column A of EleExxel templates	29
VariableTemplates in an EleExxel template file	30

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Defining the exported attributes (data fields)	32
Defining the exported data objects (instances, records)	34
Special handling of dynamic attributes in EleExxel	36
Special handling of text data fields in EleExxel	37
Formatting of data rows in EleExxel template files	37
EleExxel repeatable templates	39
What means “repeatable template”?	39
Structure of an EleExxel repeatable template	39
Discussion of a repeatable template for EleExxel	40
EleExxel non-repeatable templates	41
What means “non-repeatable template”?	41
Discussion of a non-repeatable template for EleExxel	41
More templates and sample pages on our websites	43
EleExxel projects and their settings	44
Overview and purpose of projects in EleExxel	44
EleExxel project table	44
The EleExxel project details form	45
The settings of an EleExxel project	46
The settings of an EleExxel project - handling of page titles	49
The settings of an EleExxel project - handling of rows and columns	49
The settings of an EleExxel project - handling of auto-fit	50
EleExxel tips and tricks	52
Use of the print area	52
Calculating prices in and by template files	52
Using cell references	52
Create a non-repeatable template from the output of a repeatable template	52
Avoid language dependent text in EleExxel template files	52
Interaction problems MSOffice with EleFamily	54
Important “don’ts” when using EleFamily with MSOffice	54
Interaction between EleFamily and third party software	54
ELEPRICELIST	55
Overview on ElePricelist	55
ElePricelist brief abstract	55
How ElePricelist is used	56
ElePricelist templates – structure, format, data references	57
What are ElePricelist templates?	57
References to data in ElePricelist templates	57
Defining the exported attributes (data fields) in ElePricelist	58
Defining the exported data objects (instances, records) in ElePricelist	58
Special handling of dynamic attributes in ElePricelist	60
ElePricelist repeatable templates	61
Overview	61
What means “repeatable template”?	61
Structure of an ElePricelist repeatable template	61
Sample output of a repeatable template for ElePricelist	63
ElePricelist non-repeatable templates	65
What means “non-repeatable template”?	65

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Discussion of a non-repeatable template for ElePricelist	65
More templates and sample pages on our websites	68
ElePricelist embeds EleExel and ElePricelist projects.....	69
Overview	69
Syntax for embedding other projects in an ElePricelist master project.....	69
Structure of an ElePricelist embedded EleExel template	70
ElePricelist projects and their settings.....	71
Overview and purpose of projects in ElePricelist	71
ElePricelist project table.....	71
The ElePricelist project details form	72
The settings of an ElePricelist project	73
The settings of an ElePricelist project – page breaks	74
The settings of an ElePricelist project – handling of formatting	75
The settings of an ElePricelist project – handling of auto-fit	75
ELEPRODUCTSITE	76
Overview	76
The ideas behind EleProductSite.....	76
The most important components of EleProductSite	78
Static HTML pages versus dynamically generated HTML	78
The website grouping system	79
Overview	79
The structure of the website grouping system	79
Data fields of the website grouping system.....	81
The website grouping system form	82
Menu in the tree view of the website grouping system	83
Data fields related to the root of the website grouping system.....	84
Data fields related to the branches of the website grouping system.....	86
Data fields related to the pages of the website grouping system	88
EleProductSite projects and their settings.....	89
Overview and purpose of projects in EleProductSite.....	89
EleProductSite project table	89
EleProductSite project details form.....	89
The settings of an EleProductSite project	90
The EleProductSite generator.....	91
Overview	91
Starting the EleProductSite generator	91
Executing procedures linked to a website grouping level	91
Available website grouping level types.....	91
Exporting an Elepub publication to EleProductSite.....	93
Overview	93
Structural differences between an Elepub publication and a product website	93
System settings for EleProductSite.....	95
The EleProductSite related form in system settings	95
Error messages in EleProductSite	97

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Overview on EleOffice and EleFamily

EleFamily

EleFamily consists of two major products, Elestore and Elepub, and of a series of smaller and additional products such as EleExcel, ElePricelist and so forth. For the two flagship products Elestore and Elepub there are separate and very comprehensive user manuals. Most of the other EleFamily products are explained in this user manual.

In many cases you will also have to refer to the Elestore user manual, because most of the EleFamily products use Elestore and its multi-media database as their technical fundament.

Stand-alone publishing products and extensions to Elestore

EleExcel, ElePricelist, EleProductSite and Elequote are optional software modules for Elestore as well as autonomous software products, which in this case always includes a basic version of Elestore. This is the reason why this manual primarily discusses the use of these software products from the perspective of Elestore.

The general goals of these modules are:

- To make as much and as flexible use of the content in Elestore as possible
- To provide simple to use, yet flexible entry-level publishing software products
- To create products, which can be utilized by any technically rather unskilled PC user knowing MS Office software
- As well as to create additional extensions to the full and comprehensive Elestore software

Here is a short overview on the EleFamily products. Most of them, when used as stand-alone products, include a basic version of Elestore.

Elestore

Elestore is the Product Content Management system with multi-media database and therefore the central and most important product of the entire family. It stores and administrates any type of product related data and catalog content and acts as the master database providing all other members of the EleFamily with the needed data.

EleOffice

Actually, EleOffice is not a single product but the term is rather used to cover all those members of EleFamily, which make intensive use of the MSOffice software products with the exception of Elestore. Therefore, the term EleOffice currently covers EleRobot, EleCalc, EleExcel, ElePricelist, EleQuote and EleProductSite. Note that Elestore is typically not covered by the term EleOffice although Elestore also makes intensive use of MSExcel and MSWord.

EleExchange

EleExchange enables Elestore to exchange product data with customers, suppliers and other business partners. EleExchange supports most important data standards for the transmission of product and price data like Datanorm, BMEcat, XML etc.

EleExcel

EleExcel provides complex multi-line exporting, formatting and printing capabilities for article objects to MSExcel using MSExcel template sheets. These template sheets can be created easily by any PC user with some experience in MSExcel.

ElePricelist

ElePricelist uses the more flexible and more comprehensive formatting capabilities of MSWord. It constitutes a quick and stand-alone price list generator, HTML generator, printing and exporting software. ElePricelist can also embed several MSExcel sheets generated by EleExcel into one MSWord document, because EleExcel is part of ElePricelist.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleProductSite

EleProductSite automatically generates entire websites in HTML with detailed product pages and with the navigation logic from content in Elestore and/or from HTML pages from other sources. EleProductSite always comprises of EleExxel and ElePricelist so that EleProductSite not only generates product websites but also supplies all of the output and results of EleExxel and ElePricelist.

The layout of the generated HTML product pages can be freely and easily designed by any user of MS Office software by defining template files in MSExcel or MSWord. This is a very simple yet extremely effective solution for every normal user of MS Office without any need to know the internals of HTML.

Additionally and alternatively, EleProductSite can also use HTML template files, which are filled with data from Elestore. This is the professional solution for the experienced HTML designer offering total design flexibility and control over the generated HTML pages. In both cases the navigation as well as the conversion of article numbers and page number to hyperlinks is generated by EleProductSite. It also delivers the HTML pages for use by the EleAIS webserver.

EleProductSite is capable of merging HTML pages from four different sources:

- HTML pages generated by EleProductSite via EleExxel and/or ElePricelist
- HTML pages, which were converted from PDF by Elecat-P2H
- HTML pages generated by Elepub
- Third party HTML pages, which were created by some other software or HTML editor and which are filled with data by EleProductSite.

EleProductSite merges all HTML from different sources into one concise tree of HTML pages and it also creates the bi-directional links to and from the EleAIS database. As a consequence, these pages can be opened from within article details view with database data as presented by the EleAIS webserver.

This combines page views of the data with database views via bi-directional linking created by EleProductSite.

EleAIS

EleAIS, the Article Information System, is a webserver application for publishing content from Elestore on the Internet for use in any Internet browser. EleAIS consists of dynamically generated product catalog lists and database views, which are bi-directionally linked with well designed and individually formatted, detailed and paper-like product pages in HTML, which are typically created by EleProductSite.

EleRobot

EleRobot is not an autonomous product in its own rights but it is an extremely powerful module extending Elestore and several other EleFamily members. EleRobot uses MSExcel to fully automatically modify, manipulate or calculate content of data objects and it can also create new content by manipulating or calculating it from existing content.

Elecalc

Elecalc administrates and calculates sales and purchase prices using simple or complex and conditional user-defined formulas, which can be specific to customers and customer groups. Elecalc can embed and use MSExcel spreadsheets so that any user-defined calculation logic can be applied by any user with some basic knowledge of MSExcel.

Elequote

Article objects in Elestore with their extremely comprehensive set of attributes (variables) and their embedded text objects are used to create individually formatted and expressive multi-page quotes in MSWord based on article sequences imported from an MSExcel sheet or from orders. Order data is either imported or entered manually in the Elequote order entry software. Elequote uses predefined templates in MSWord files to format data content from Elestore.

Elepub

Elepub is the Object Database Publishing system, which generates attractive and flexible layout, pages and entire publications fully automatically from data in Elestore for paper, e-paper and electronic media. Elepub produces all electronic publications autonomously without any need for DTP software but it also very closely co-operates with DTP software.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleHelp

EleHelp is the general help tool and help reader, which is fully integrated in the EleFamily software products. EleHelp presents help texts, graphics and images from the help database at runtime to the user of the EleFamily software.

EleHelp is much better and much tighter integrated into the EleFamily software than MSHelp could ever be. EleHelp interprets software states and data from the database. There is also an EleHelp generator, which creates the EleHelp runtime library as well as HTML help from text objects in Elestore by using sequences for defining the exported text objects.

EleRobot

Overview

EleRobot brief abstract

EleRobot is the perfect blend of:

- the sophisticated Elestore database management system
- with MSEXcel, the most widespread calculation and data manipulation tool.

EleRobot is an extremely powerful tool and software module, which automatically modifies and calculates data objects in the Elestore database. EleRobot fully embeds and uses MSEXcel for its data calculation and data modification tasks. This is why the user of EleRobot can freely define any logic, from simple to complex, in MSEXcel template sheets, which is then applied by EleRobot on the data in the Elestore database.

Through EleRobot, all the features and capabilities of MSEXcel including functions and the embedded Visual Basic programming language can be applied on data in the Elestore database. This offers total flexibility and an extreme productivity in maintaining Elestore content in the shortest time possible.

How EleRobot automatically modifies data in Elestore

This is how EleRobot automatically changes data in the Elestore database: In a user-defined MSEXcel template sheet for EleRobot you have defined the attributes (variables, data fields), which are to be exported into a copy of this template sheet. The exported data is then processed, calculated or modified by formulas, which you have applied to the MSEXcel sheet yourself. The result is read back immediately by EleRobot either into the same attributes or into other user-defined attributes of the same data object.

This is how EleRobot processes all selected data objects one after the other and changes data in a fully automatic procedure. A data modification protocol file is generated during an EleRobot process, from which you can see how the content of your database is modified by EleRobot. Additionally, entries to the history change database are made if this is part of your Elestore installation.

Versions of EleRobot

Unlike most other EleFamily products, there exists no stand-alone EleRobot, because EleRobot is always an integrated part of (or option for) other members of EleFamily, especially of Elestore, EleExxel, EleCalc, EleQuote, ElePricelist and EleProductSite.

EleRobot projects

Overview on EleRobot projects

EleRobot uses projects, which store all relevant settings for a job that was once performed by EleRobot for a simple and quick re-use at any later moment. Projects and their settings are stored in the Elestore database.

Purpose and goals of the projects and their settings

Re-using what you or your colleagues have done before, that is the primary goal of the projects and their settings.

By storing all user defined settings in a project object in the database, you or your collaborators can easily and quickly start one of these projects without having to think about the many details that might be associated with a project.

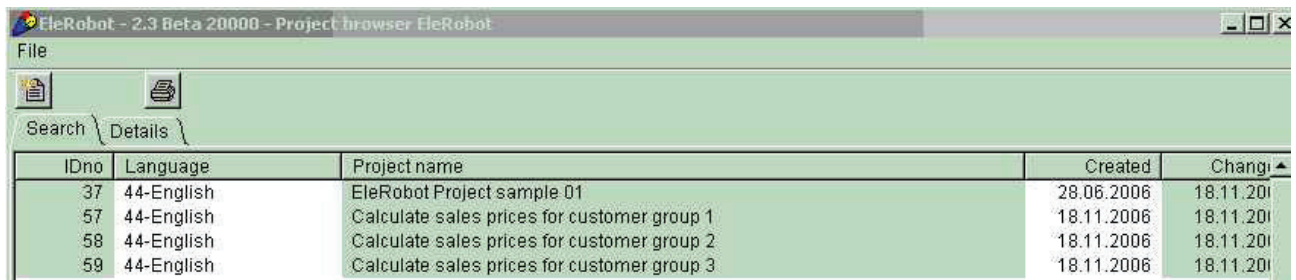
All that is typically needed is:

- 1) Select the appropriate project
- 2) Select the data objects that are to be processed
- 3) And off we go!

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleRobot project table

This is a partial view of the project selection table of EleRobot:



The screenshot shows a window titled "EleRobot - 2.3 Beta 20000 - Project browser EleRobot". It has a menu bar with "File" and a toolbar with icons for file operations. Below the toolbar are two tabs: "Search" and "Details". The "Details" tab is active, displaying a table with the following data:

IDno	Language	Project name	Created	Change
37	44-English	EleRobot Project sample 01	28.06.2006	18.11.2006
57	44-English	Calculate sales prices for customer group 1	18.11.2006	18.11.2006
58	44-English	Calculate sales prices for customer group 2	18.11.2006	18.11.2006
59	44-English	Calculate sales prices for customer group 3	18.11.2006	18.11.2006

You can select any EleRobot project from this table for simple re-use. Click on one of the existing projects and then switch to the details form to re-use the stored project with or without modification of the settings.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

The EleRobot project details form

View of the EleRobot project details form

EleRobot - 2.3 Beta 20000 - EQUStefApplicationDriverForEleRobot

File

Search Details

Settings

Project name: EleRobot Project sample 01

Name of template file: .\EleRobot\EleRobotTemplate.xls ... ☒ Use inheritance

Export type: 200-Export all of a grouping level

Office product group: 1-:Second level office grouping system ...

Language: 44-English

Import settings

☒ Re-import data automatically

Cancel Select articles Export

This is the EleRobot project settings form.

Menu and buttons in the EleRobot project settings form

[File menu in the EleRobot project settings form](#)

The file menu contains the usual menu items for adding, deleting, saving, or printing a project.

[Button Select articles in the EleRobot project settings form](#)

Pressing this button opens the form for selecting article objects.

[Button Export in the EleRobot project settings form](#)

Pressing this button starts the processing of this project. If you have not yet selected the article objects, the form for selecting article objects will be opened first.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

The settings of an EleRobot project

Project name

The field “project name” stores a text describing the purpose of this project. You should select some meaningful explanation of what this project does so that it will be easy for you or other users to select and re-use this project without having to look at all the settings.

Name of the template file

Selecting the template file is the most important of all settings. The template file defines not only the attributes (data fields), which are to be processed, but especially how they are processed, changes, calculates etc.

Please note that there is no possibility for EleRobot to check whether you have selected a correct template file.

Please read carefully the description on how templates files are to be defined before making your own template files. In the beginning, the best is probably if you take one of the sample files, which were shipped with the software, and modify it according to your needs.

With inheritance

Provided that your version of Elestore supports data inheritance you can select or deselect data inheritance for your current project with this checkbox.

If you deselect data inheritance and your data has been stored with active inheritance the output data will most likely not correspond to what you have expected, because in these cases only the private data of child objects will be exported. On the other hand, there are occasions where this is exactly what one wants, for example when you want to control the private values of child objects.

For a more detailed discussion of data inheritance please refer to the corresponding chapter in the Elestore user documentation. Please note also that some versions of EleRobot do not support data inheritance and that in these cases you cannot activate this checkbox.

Export type

This setting defines the scope and number of data objects, which are to be processed. The data objects are searched and selected in the article selection form. In some versions of Elestore and EleRobot StoredSelections of data objects can also be used.

[Export selected objects \(110\)](#)

This setting exports only those objects, which were individually selected by the user in the article selection form. Selecting objects in this sense means that single rows or the range of rows have been selected in the search results table.

[Export single block of objects \(120\)](#)

This setting exports the current block of data objects from the search result form. Please refer to the documentation of Elestore for a more detailed explanation of what a block of objects means in this sense.

[Export all for search condition \(130\)](#)

This setting exports the entire search result as defined by the search condition in the article search form.

[Export of a grouping level \(200\)](#)

Selecting this export type activates the selection field “Office product group” in the project settings. With this export type EleRobot will traverse all article objects, which are linked to the selected office product grouping level and its sub-levels and EleRobot will use these article objects during the generation process.

With this export type you can use the office grouping system for defining the scope and the sequence of article objects, which are processed by EleRobot.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Office product group

This field is only available if you have selected in the field type of template the type 200, that is export all of a grouping level. In all other cases this field is disabled.

This field allows you to select a grouping level from the office grouping system (please note that there are typically several different grouping systems available). By selecting a specific grouping level you tell EleRobot, which article objects to process in which sequence, which is both defined by the grouping level and its article objects.

For further details please refer to the chapter about the office grouping system.

Language

Elestore and all other EleFamily products support multi-lingual content, of course, because the world is and will always be multi-lingual.

In this selection field you decide about the language, which is to be used for all language dependent data fields such as strings and text fields. This setting allows you to use one template file for any number of languages provided that all the language dependent content comes from the Elestore database.

Re-import data automatically

EleRobot will re-import the data resulting from the calculation of the template sheet only if this checkbox is activated. If this checkbox is deactivated only a protocol file on the data changes is created and the content of the database is not modified.

The purpose of the activating this checkbox is in the fact that you can have a test run, which will not modify the content of your database. Instead, only a data modification protocol file will be generated, from which you can see how the content of your database would have been modified by EleRobot when using the current settings and the selected template file together with this checkbox activated.

See also the chapter about the EleRobot data change protocol file.

Running EleRobot

How to run EleRobot

This is very simple: Start the EleRobot project manager from the main menu -> applications -> EleRobot. This will open the EleRobot project form and present the form for selecting EleRobot projects.

Use an existing project

If you want to use an already existing project then select one via a double-click or selected one with a single click and then press the details tab.

Create a new project

If you want to create a new project then select from the main menu of the EleRobot browser -> New and fill out all of the project settings.

Prepare all settings

Take care to define all settings according to what you want to do. Especially the selection of the correct EleRobot template file in MSExcel format is essential. If you don't yet know how to create your own MSExcel template file for EleRobot then simply use one of the sample files delivered with the installation. You can also modify one of these sample template files by using MSExcel before starting EleRobot.

Start EleRobot

If all of your settings are correct you just press the button ->Export and this will start EleRobot, which immediately processes all selected objects in the selected EleRobot template sheet one after the other. It applies the calculation logic con-

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

tained in the EleRobot template file on the attributes of every processed data object and subsequently reads the resulting data immediately back into the Elestore database.

[Data modification history](#)

Provided your Elestore installation uses the data modification history database you will find entries on all modified data entities in the history database.

Attribute naming conventions

Note that there is a difference between EleRobot and EleExxel regarding the naming conventions for attribute names: In EleRobot you are not obliged to add the hash character (“#”) to the names of attributes as it is the case in EleExxel. EleRobot treats attributes names with or without the hash character in the same way.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleRobot templates – structure, format, data references

What are EleRobot templates?

This chapter discusses the elements and the structure of EleRobot template files and how data is referenced in templates.

EleRobot uses templates in the format of MSEXcel files (*.xls). These are the blueprints, the construction plans for EleRobot to let MSEXcel calculate and modify data in Elestore. The template files tell EleRobot where to place what kind of data, they define the logic, which is to be applied on the exported data, and they also define from where the data is taken to be read back into Elestore.

The templates used by EleRobot are different from the templates used by EleExel or ElePricelist although the principles and also much of the syntax that is used to reference Elestore data are identical.

EleRobot and user-defined templates

EleRobot is based on user-defined MSEXcel template files

EleRobot applies the calculation logic contained in a user-defined MSEXcel sheet on the exported data and then EleRobot immediately re-imports the calculated or modified results from the MSEXcel sheet back into the Elestore database.

Sample EleRobot template files are shipped with the software

Part of a software installation with EleRobot is a couple of sample template files (*.xls, the MSEXcel spreadsheet format) that you can use out of the box together with the content of the Elestore database. This is also the case for the trial version.

We recommend you to read this manual as well as to look at these sample files. The best way to create your own template files is probably by taking one of the samples and modify it according to your requirements and ideas. This is presumably much better and simpler than creating your own template files from scratch.

Of course, our support will be pleased to assist you in defining your own very individual template files and additionally there is a library of sample files available for purchase at very low costs.

The structure of an EleRobot template file

Named areas in an EleRobot template file

In MSEXcel one can give areas of cells a name. This feature is used by EleRobot to define the area, into which EleRobot is to export data, as well as the area, from where EleRobot re-imports the calculated results back into Elestore.

In fact, in order to work properly an EleRobot template sheet **must have** the two named areas as described below.

It is extremely important that these two named areas do exist in an EleRobot template file or otherwise there will be an error from MSEXcel that our software cannot intercept. We have not found any way to prevent this error from crashing the application and this is a problem of the interface to MSEXcel, which was written by Microsoft and which is therefore beyond our control.

Two named areas of cells

An EleRobot template sheet must contain two named areas:

- The area “toExcel” is the area of cells, where EleRobot exports the data values from Elestore so that they can be processed by the formulas of the EleRobot template sheet.
- The area “fromExcel” is the area of cells, from where EleRobot reimports the calculated results back into Elestore.

Both area names do not contain any quotes and they must be named exactly as shown above inside the quotes.

Two columns of cells are expected in both named areas

It is important that each of these areas consists of two directly neighboring columns:

- The left column must contain the attribute names, which are to be processed.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

- The right column must contain the values, which are either exported to MSEXcel or from where the values are reimported from MSEXcel into Elestore.

This is true in the same way for both named areas.

Other cell areas of an EleRobot template spreadsheet outside these two named areas can be used as you like. They will be disregarded by EleRobot. You can use them for comments or explanations to human users of the EleRobot template file. In the sample1 this is the case for column A.

The types of references and data in cells in EleRobot templates

Overview on the purpose of referencing content in cells in EleRobot template files

In general, much of the universal functionality of spreadsheet software comes from referencing other cells and this is a frequent practice in spreadsheets. This powerful feature of spreadsheet software is also used by EleRobot, of course, but with one essential difference: Most of the cell references in EleRobot templates refer to data in the Elestore database.

This is why EleRobot extends the capabilities of MSEXcel by those of Elestore and vice versa.

Cells of an EleRobot spreadsheet reference attributes (data fields, instance variables) of data objects. The identification of the data objects, which EleRobot is supposed to process, is defined in the project by the user via one of several forms of data selections.

Referencing data attributes (fields) in EleRobot template files

Every cell in an EleRobot template file, which is to be filled by EleRobot with some data from the Elestore database, must contain a reference to an attribute of the data object class in Elestore. Please note that we are using the term “attribute” while the terms “data field” or “variable” are also frequently used. In most cases they can be used synonymously.

The attribute, which is referenced in a cell, is identified by the attribute name. This is the same name that is used in the Elestore database. Please look into the Elestore user manual to understand how attributes are named in Elestore. We use a very simple but concise and obvious naming convention.

Attributes in EleRobot templates are neither referenced directly nor indirectly like in EleExcel. Instead, there are specific columns, which must contain the attribute names, and other columns, which contain the values.

Use of several rows

Of course, you can use several rows of the EleRobot template file, each of which refers to a different attribute. Like this you can process several or many attributes in one run. See the samples.

One attribute should be contained only once in a template file

You should be aware that one attribute should only be contained once in the “toExcel” area of one EleRobot template file. Especially, you should be aware that EleRobot first exports all values for all attributes, which are referenced in a template sheet, to a copy of this template sheet. After this export EleRobot tells MSEXcel to perform its calculation operations on the data and it then reads back all attributes from the result column.

As a consequence this means that all attributes are first exported, then calculated and then re-imported. In other words, you cannot expect that an intermediate result, which was calculated in let’s say row five, will first be read back by into Elestore and will then be exported again to a lower row. This is not the case. In order to achieve this you would have to run two separate processes one after the other, which both refer to the same attribute.

One can use one variable in more than one row but in such cases one has to understand that all of the data is read back from Excel and written to the database in one pass.

Object identifications in the EleRobot template file

There are never any object identifications such as article numbers contained in an EleRobot template sheet. In this respect an EleRobot template can be compared to a repeatable EleExcel template file.

The the data objects, which are to be processed by EleRobot, are defined by the selections that a user makes.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

The EleRobot template files only contain identifications of the attributes, from where data is to be exported into the MSeExcel file, and identifications of the attributes, into which the result data is read back into Elestore.

Placing of the data modification formulas

There are essentially two possibilities how you can apply your own logic and where you can store your formulas in EleRobot template files that are to be applied by EleRobot via MSeExcel on the exported data values:

- You either place these formulas directly into the cells of the data column in the “toExcel” area.
- Or you let the cells in the data column in the “toExcel” area refer to some other columns outside the two named areas where you place your formulas as you like. Like this you can even apply multi-step calculations, which are often easier to enter and to edit in the cells of MSeExcel than complex formulas in one single cell.

In any case, the cells of the data column in the “toExcel” area must contain the result, which is to be read back into Elestore.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleRobot template samples explained

Explanations in EleRobot template files

Part of the shipped software is a couple of EleRobot template files, which are ready for immediate use. They are also linked to EleRobot projects which come with the installed database (at least in the trial version).

If you look into these EleRobot template files you will see additional sheets in English and Deutsch beyond the actual calculation sheet, which explain the content of the actual EleRobot calculation sheet (typically the first in the file).

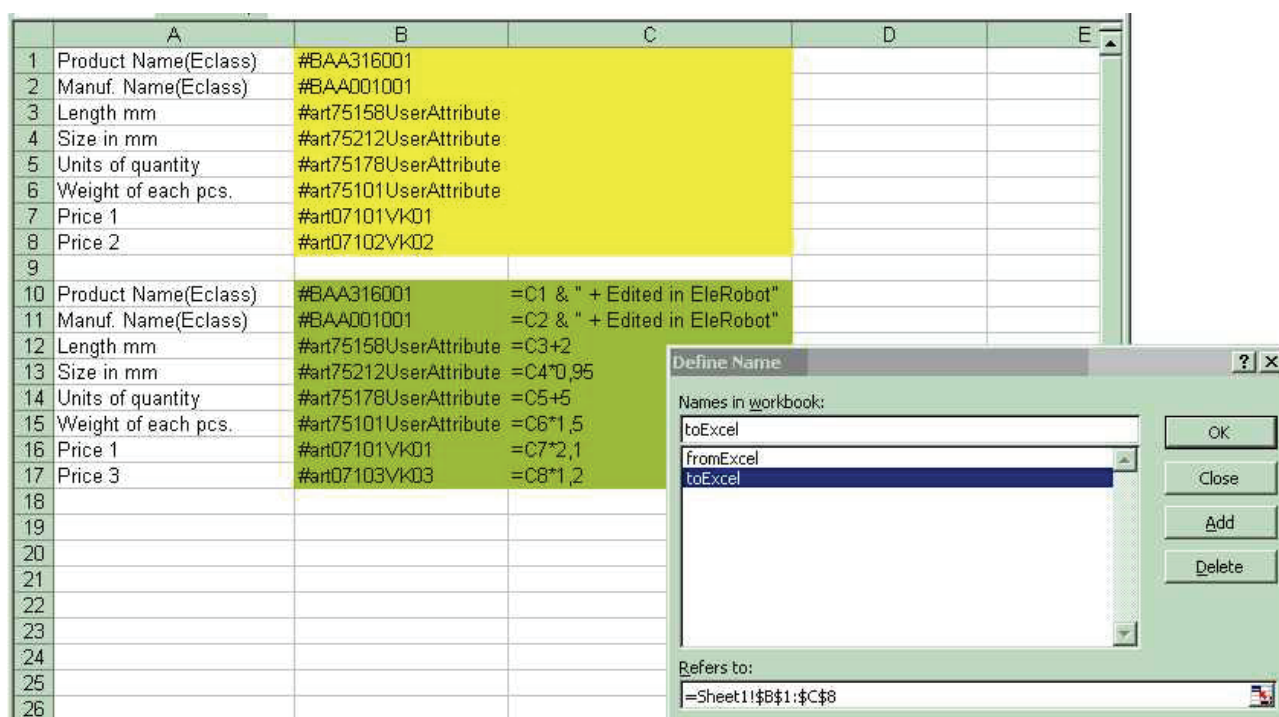
Template files on product websites

You also find additional template files on the EleOffice product websites. We recommend you to use these sample files for constructing your own very individual template files.

A EleRobot template sample

View of EleRobot template sample

This is a screenshot of some EleRobot template sample file with the overlaying form for naming areas:



This screenshot has been taken with the area definition dialogue form opened and placed over the EleRobot template sheet. For your convenience and to make it better visible the two areas have been coloured:

- The yellow area is the “toExcel” area
- And the green area is the “fromExcel” area.

The colours are not needed in real life.

Note also that this screenshot was made with the MSExcel setting “Show formulas” in the Options form activated, so that the formulas contained in cells C10:C17 are visible. This is only for the screenshot and also not needed in real life.

The column A of this example contains user information, which is disregarded by EleRobot, because this data is outside the two named areas.

There is no need to place these cell areas as they are placed in this sample1 as you can see from other EleRobot samples.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Area named “toExcel” in an EleRobot template file

This is the area where EleExcel stores the data that it exported from Elestore. This area must consist of two neighboring columns, the left one for the attribute names and the right one for the data. Column B of this area in the above example tells EleRobot, which attributes to read from every processed Elestore article object. The values of these attributes are stored in column C in the same row of the attributes definition.

Area named “fromExcel” in an EleRobot template file

This is the area from where EleRobot reads the modified or calculated data to store it back into Elestore. This area must consist of two neighboring columns, the left one for the attribute names and the right one for the data. The left column of this area tells EleRobot, into which attributes it has to store the values of the right column. Please note that in this sample the right columns are equipped with formulas, which refer to cells of the area “toExcel”.

Other areas in an EleRobot template file

All other areas outside these two named areas are ignored by EleRobot. You can therefore place documentation and explanation to human readers in an EleRobot template sheet as long as this is outside the two needed named areas.

Discussion of some formulas in the EleRobot template sample1

In order to help you to better understand what is going on in this EleRobot sample template here is an explanation on some of the formulas in column C:

- The formula in cell C10 takes the value from cell C1 and adds the text, which is enclosed in quotes. The same happens in the formula in cell C11.
- The formula in C12 uses the value from cell C3 and adds an integer of 2.
- A similar thing happens in C13, which takes the value of cells C4 and multiplies it by 0,95. Similar calculations are done in the other formulas in column C.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Another EleRobot template sample

View of EleRobot template sample2

A screenshot of EleRobot template sample2 with the named areas coloured (only for your better reading):

	A	B	C	D
1	Product Name(Eclass)	#BAA316001		
2	Manuf. Name(Eclass)	#BAA001001		
3	Length mm	#art75158UserAttribute		
4	Size in mm	#art75212UserAttribute		
5	Units of quantity	#art75178UserAttribute		
6	Weight of each pcs.	#art75101UserAttribute		
7	Price 1	#art07101VK01		
8	Price 2	#art07102VK02		
9				
10	Product Name(Eclass)	#BAA316001	=C1 & " + Edited in EleRobot"	
11	Manuf. Name(Eclass)	#BAA001001	=C2 & " + Edited in EleRobot"	
12	Length mm	#art75158UserAttribute	=C3+2	
13	Size in mm	#art75212UserAttribute	=C4*0,95	
14	Units of quantity	#art75178UserAttribute	=C5+5	
15	Weight of each pcs.	#art75101UserAttribute	=C6*1,5	
16	Price 1	#art07101VK01	=C7*2,1	
17	Price 3	#art07103VK03	=C8*1,2	

For your convenience and to make it better visible the two areas have been coloured:

- The yellow area is the “toExcel” area
- And the green area is the “fromExcel” area.

The colours are not needed in real life.

Note also that this screenshot was made with the MSExcel setting “Show formulas” in the Options form activated, so that the formulas contained in cells D1:D8 are visible. This is only for the screenshot and also not needed in real life.

There is no need to place these cell areas as they are placed in this sample2 as you can see from the EleRobot sample1.

Note that the first release of EleRobot contains a small inconsistency versus EleExxel in that the names of attributes must NOT contain the hash character (“#”) like attributes in EleExxel. This will be changed ASAP so that EleRobot uses the same attribute naming convention as EleExxel.

[Column A of the EleRobot template file](#)

Column A of the EleRobot template file contains the names of the attributes, from where data is to be taken and to be exported into the next column B of the EleRobot template file. In column A you place the names of the attributes. You find details in the user manual on EleExxel.

[Column B of the EleRobot template file](#)

Column B of the EleRobot template file must be left empty, because this is where EleRobot stores the values, which are exported from the attributes as defined in column A of the same row. Your formulas, which calculate or manipulate the exported data, must refer to this column B in order to be able to use and process the exported data.

[Column C of the EleRobot template file](#)

Column C contains the names of the attributes, into which EleRobot reimport the result data that is contained in column D.

The names of the attributes in column C can be different from the names of the attributes in column A. This means that you can import the results of your data calculations or manipulations also into another attributes as those from where the original data was originally exported.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

[Column D of the EleRobot template file](#)

Column D contains the results of your calculations, which is taken by EleRobot and read back into the Elestore database into the attributes, which are defined a column C. Column D is also the one, where the formulas should be contained, which calculate or manipulate the data from column B. Alternatively, the cells in column D point to some other cells from where the calculated result data is taken.

Please refer to the next chapter about the placing of formulas.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleRobot data change protocol file

Overview and purpose

EleRobot is extremely powerful in automatically modifying your database content but this can also be quite dangerous, of course, if you make a mistake.

Due to the complexity of the operations an undo over such database operations is not feasible.

As a consequence, this incorporates the risk to easily jeopardise the content of your database if you make a severe mistake for example by selecting the wrong EleRobot template file or by having a wrong formula inside your template. Somehow this risk is the other side of the coin of the great comfort and productivity of EleRobot.

In order to minimise this risk we have therefore included two features, which help you to avoid jeopardising your database:

- The first is a switch in the EleRobot project settings that disables the re-import of modified data into the database. This switch makes only sense, of course, together with the second feature, the protocol file.
- The second and most important feature is a protocol file, which is created during an EleRobot process run and which reports all data modifications made by EleRobot to the content of your database.

This protocol file together with deactivating an immediate re-import of the modified data into your database are a simple and efficient ways to have a test run performed on your database content without risking any unwanted data modification to this content.

You can therefore first make a test run. You should then thoroughly check the results in the protocol file and you should only start a real process if these results are according to your expectations.

In any case the first commandment of every computer user does apply: “Thou shall make backups regularly!”

Sample of an EleRobot data change protocol file

Here is a part of a sample data change protocol file, which was created when running the above mentioned EleRobot sample template. Of course, only the first few lines are reproduced here.

Partial reproduction of a protocol file:

Time stamp Montag, 20. November 2006 8:08:26.000

Name of template file E:\EleRobot\Templates\EleRobot-894.xls

Import condition 110-Export selected objects

Language English

Re-import data automatically true

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 5,5';resultSourceAttribute' ->
#BAA316001;resultSourceValue' -> 'Single Open-Ended Spanner';resultDestinationAttribute' ->
#BAA316001;resultDestinationValue' -> 'Single Open-Ended Spanner + Edited in EleRobot' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 5,5';resultSourceAttribute' ->
#BAA001001;resultSourceValue' -> 'Elora';resultDestinationAttribute' -> #BAA001001;resultDestinationValue' -> 'Elora +
Edited in EleRobot' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 5,5';resultSourceAttribute' ->
#art75158UserAttribute;resultSourceValue' -> '75';resultDestinationAttribute' ->
#art75158UserAttribute;resultDestinationValue' -> '77' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 5,5';resultSourceAttribute' ->
#art7512UserAttribute;resultSourceValue' -> '5,5';resultDestinationAttribute' ->
#art7512UserAttribute;resultDestinationValue' -> '5,225' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 5,5';resultSourceAttribute' ->
#art75178UserAttribute;resultSourceValue' -> '10';resultDestinationAttribute' ->
#art75178UserAttribute;resultDestinationValue' -> '15' ]
```

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 5,5';resultSourceAttribute' ->
#art75101UserAttribute;'resultSourceValue' -> '8';resultDestinationAttribute' ->
#art75101UserAttribute;'resultDestinationValue' -> '12' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 5,5';resultSourceAttribute' ->
#art07101VK01;'resultSourceValue' -> '3,430000067';resultDestinationAttribute' ->
#art07101VK01;'resultDestinationValue' -> '7,20300014' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 5,5';resultSourceAttribute' ->
#art07102VK02;'resultSourceValue' -> '2,74000001';resultDestinationAttribute' -> #art07103VK03;'resultDestinationValue'
-> '3,288000011' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 6';resultSourceAttribute' ->
#BAA316001;'resultSourceValue' -> 'Single Open-Ended Spanner';resultDestinationAttribute' ->
#BAA316001;'resultDestinationValue' -> 'Single Open-Ended Spanner + Edited in EleRobot' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 6';resultSourceAttribute' ->
#BAA001001;'resultSourceValue' -> 'Elora';resultDestinationAttribute' -> #BAA001001;'resultDestinationValue' -> 'Elora +
Edited in EleRobot' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 6';resultSourceAttribute' ->
#art75158UserAttribute;'resultSourceValue' -> '75';resultDestinationAttribute' ->
#art75158UserAttribute;'resultDestinationValue' -> '77' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 6';resultSourceAttribute' ->
#art75212UserAttribute;'resultSourceValue' -> '6';resultDestinationAttribute' ->
#art75212UserAttribute;'resultDestinationValue' -> '5,7' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 6';resultSourceAttribute' ->
#art75178UserAttribute;'resultSourceValue' -> '10';resultDestinationAttribute' ->
#art75178UserAttribute;'resultDestinationValue' -> '15' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 6';resultSourceAttribute' ->
#art75101UserAttribute;'resultSourceValue' -> '8';resultDestinationAttribute' ->
#art75101UserAttribute;'resultDestinationValue' -> '12' ]
```

```
Result[true] - resultTrueEleRobotImport - Result[ 'Article number' -> '894- 6';resultSourceAttribute' ->
#art07101VK01;'resultSourceValue' -> '1,429999948';resultDestinationAttribute' ->
#art07101VK01;'resultDestinationValue' -> '3,00299989' ]
```

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleExcel

Overview on EleExcel

EleExcel brief abstract

EleExcel allows the export of data from Elestore to well-formatted MSeExcel sheets, i.e. simple database publishing. Its practical usage is:

- Generates well formatted and printable MSeExcel sheets from Elestore content as a simple form of database publishing.
- Generates product lists in HTML with links to other HTML product pages and/or to the EleAIS database.

The EleExcel software is available in different forms:

- As a stand-alone EleExcel software product incorporating a basic version of the Elestore content database
- As an optional module extending Elestore
- As a combined product together with Elecalc, Elequote and/or other members of the EleFamily
- As an integral part of ElePricelist or EleProductSite, which both include EleExcel.

The Elestore user manual is needed as an extension of this EleExcel manual.

EleExcel makes the most of MSeExcel

MSeExcel has excellent formatting and printing as well as very comprehensive exporting capabilities with support for many different data formats. Almost every computer user knows how to use MSeExcel and this software is available on most computers anyway.

This is why EleExcel provides comfortable and comprehensive mechanisms for using MSeExcel and its features as a generator for the data stored inside the Elestore database. EleExcel fully automatically uses and drives MSeExcel as an output and print generator as well as an export manager supporting most physical data formats in use today.

The standard Elestore software already fully embeds MSeExcel both as a QuickEditing tool as well as an export target for data from the Elestore database. This has been available since one of the first Elestore releases way back in early 2004.

EleExcel extends this by providing even far more comprehensive capabilities in using MSeExcel especially for formatting multi-line data output from the Elestore database. EleExcel provides simple, quick, yet very powerful features for using MSeExcel as a universal output and print generator and a product and price list generator for data in Elestore.

Data object types processed by EleExcel

In the current version, EleExcel processes article objects and address objects (which is limited to a special version). When integrated into Elequote, the logic of EleExcel also generates quotes and other order related output.

Beyond this we are planning to extend the capabilities of EleExcel in the future to additional data types.

How EleExcel is used

EleExcel always uses the content of an Elestore database but this is not the subject of this user manual. Please refer to the user manual on Elestore. Here we cover only how the data in Elestore is formatted by EleExcel and how EleExcel creates output.

The three steps of starting an EleExcel export process

The EleExcel export is started and performed in three simple and quick steps:

- You select the data objects for the export.
- You select a stored export project re-using previously defined export parameters and settings.
- You select an MSeExcel template file.

That is all!

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

[Select the data objects for an EleExcel export](#)

The data objects for an export are selected in the search form of the article browser. Alternatively, a StoredSelection can be used in some version of Elestore. For details please refer to the very comprehensive user manual on Elestore.

[Select the MSeExcel file to be used as template for an EleExcel export](#)

The selected export project normally also contains the name of the MSeExcel file, which is used as the template file. You can overwrite this for every single export procedure by simply selecting a different MSeExcel file as the template file.

EleExcel is very simple to use with basic MSeExcel knowledge

This are all steps needed to generate well formatted data output with EleExcel. Therefore, EleExcel is very well suitable to be used by product managers or by any normal PC user with some basic knowledge of MSeExcel.

Versions of EleExcel

EleExcel as a stand-alone product or integrated into Elestore

EleExcel is available in different variants: either as a stand-alone product consisting of a basic Elestore version or as an optional module enhancing the full Elestore software.

EleExcel as part of other products like ElePricelist and EleProductSite

EleExcel is also an integrated part of several other members of the EleFamily product series. Especially, ElePricelist and EleProductSite include a full version of EleExcel. This offers much more flexibility and more comprehensive features to these EleFamily products.

Just a few examples:

[ElePricelist makes recursive use of EleExcel](#)

EleExcel can be used within ElePricelist to generate single or multiple pages of articles objects. These MSeExcel sheets generated by EleExcel are then included into the MSWord output document, which is generated by ElePricelist. ElePricelist calls EleExcel one or several times to execute EleExcel projects and it adds the generated MSeExcel sheets into its output MSWord document.

This is what we call a recursive use of EleExcel projects by ElePricelist. For further details please refer to the description and user manual of ElePricelist.

[EleProductSite uses EleExcel to generate HTML pages](#)

There are several ways how EleProductSite makes use of EleExcel for generating HTML pages from article objects. These HTML pages are then included into the data, which is generated and exported by EleProductSite. EleProductSite creates the entire content for page-based electronic product catalogues consisting of individually designed product pages in HTML including the navigation to and between these pages.

Such pages and entire product websites are created by EleProductSite with the help and recursive use of EleExcel and/or ElePricelist. For further details please refer to the description and user manual of EleProductSite.

[Elequote uses EleExcel for detailed price information](#)

Elequote can make use of EleExcel for generating detailed price information and summaries for offers, which are created and generated by Elequote. Besides this, Elequote also generates “quotes catalogues”, i.e. special and very details product pages for the offered products like an offer-specific product catalog. For further details please refer to the description and user manual of Elequote.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleExxel's three types of templates - overview

EleExxel's three types of templates and operations

There are three fundamentally different types of templates and forms of output, which can be generated by EleExxel:

- Repeatable templates
- Non-repeatable templates
- Single page templates (templates defined by the article objects)

Please note that ElePricelist is capable of mixing repeatable templates and non-repeatable templates into one MSWord document.

EleExxel with repeatable templates

One repeatable template sheet is duplicated by EleExxel and exported into the output file as many times as there are data objects to process. All data objects are visualized in the same layout defined by this one template file.

All columns are identical for all data objects. The rows are also identically although the number of rows per data object can vary in cases where content data is or is not available.

Purpose of repeatable templates

Repeatable templates are ideal for automatically generating uniform layout with a large number of data objects. It is very simple and quick to create repeatable templates for publishing and formatting of a large number of data objects.

EleExxel with non-repeatable template pages

In a non-repeatable template all of the pages of the output file already exist preformatted in the template file but yet without content. The pages of the template file are filled with data from the data objects, which are referenced in the template file. No automatic duplication of the template is performed. The template is just filled with data.

Purpose of non-repeatable templates

Non-repeatable templates are suitable for creating much more individual and flexible layout and formatting structures, because every data object (article etc) can have its own and very individual layout. In repeatable templates all published data objects share the same layout. Of course, it takes a bit more time to define non-repeatable templates.

EleExxel with single page templates

The third type of EleExxel template sheet allows generating one page for every data object with or without its children. This template type is primarily used for generating HTML pages and product websites from Elestore content.

Another specialty is in the fact that the templates used are stored with references to article groups or article objects so that they all could have their own individual templates. Therefore, very different layouts can be used to visualize the article objects and theoretically every single data object (article) can have its own and very individual layout template.

Purpose of single page templates

This template type is ideally suitable for creating details pages in HTML. For example, one can use the output of a repeatable template to create a product list as an overview with article numbers as links to individual details pages, which are created from page templates. This is one way how EleProductSite generates an entire website from Elestore content.

Single page templates are only part of EleExxel if it is embedded in EleProductSite.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

User-defined templates in EleExxel

EleExxel is based on user-defined MSeExcel template files

EleExxel takes user-defined MSeExcel spreadsheet file as templates for creating output by filling these MSeExcel files with data from Elestore. This is the simple yet very powerful general principle of EleExxel.

EleExxel automatically merges data from its underlying Elestore database into the cells of the user-defined MSeExcel template spreadsheet file, which acts as a template and which is duplicated into an MSeExcel output file.

Sample EleExxel template files are shipped with the software

Part of an EleExxel installation is a couple of sample template files (*.xls, the MSeExcel spreadsheet format) that you can use out of the box together with the content of the Elestore database. This is also the case for the trial version.

We recommend you to read this manual as well as to look at these sample files. The best way to create your own template files is probably by taking one of the samples and modifying it according to your requirements and ideas. This is presumably much better and simpler than creating your own template files from scratch.

Of course, our support will be pleased to assist you in defining your own very individual template files and additionally there is a library of sample files available for purchase at very low costs.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleExcel templates – structure and data references

What are EleExcel templates?

This chapter discusses the elements and the structure of EleExcel template files and how data is referenced in templates.

EleExcel uses templates in the format of MSEXcel files (*.xls). These templates are the blueprints, the construction plans for EleExcel to generate MSEXcel output files. They tell EleExcel where to place what kind of data and they also define the format of the cells in the output file.

The MSEXcel files, which are created by EleExcel, are a copy and an amalgam of the original template file filled with data from the Elestore database.

The types of references and data in EleExcel template cells

Overview on referencing content in cells in EleExcel template files

In general, much of the universal functionality of spreadsheet software comes from referencing other cells and this is a frequent practice in spreadsheets. This powerful feature of spreadsheet software is also used by EleExcel, of course,

In this respect, EleExcel enhances MSEXcel in that it adds three even more powerful features to MSEXcel:

- The ability to link data in the Elestore database
- To have MSEXcel operate under full control of another application software (EleExcel)
- To perform these operations on a series of data objects in the database

Most of the cell references in EleExcel templates refer to data in the Elestore database.

Cells of an EleExcel spreadsheet reference attributes (data fields, instance variables) of data objects and often they also contain references to the data objects themselves from which EleExcel is supposed to take the values, i.e. the key values to identify data in the database.

It depends on the type of your template file if only the references to attributes are needed or if you also have to supply references to data objects. This is discussed when explaining repeatable and non-repeatable EleExcel templates.

Therefore, cells of an EleExcel template file can also reference other cells. This is often very useful to address cells in other columns, which are located outside the print area of the output MSEXcel file, which is designated for printing. Typically, the cells referencing other cells are located inside the print area while the referenced cells are located outside the print area. This, too, is discussed in more detail in the chapter about cell references.

Static and variable data in an EleExcel template

An EleExcel template file consists of cells with static and with variable data. Neither of both is mandatory but a template file without variable data would not make much sense and in almost all cases at least some static data is also contained in a template file.

Static data

Static data is just some text, which is copied to the output file without being changed. Typically, this are titles or some other text, which is to appear in the generated output file.

Variable data

Variable data in a template file consists of references to the Elestore database. This variable data is replaced by EleExcel during the generation run with the data from the database.

Static data in cells of an EleExcel template

Of course, cells in an EleExcel template file can also contain static data, which is copied by EleExcel to the output file and which is repeated with every new instance that is created from this template, when using a repeatable template file.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

The best example is looking at the title rows, which constitute a special form of static output. Additionally, you can place any static data directly in any cell of the template file in any data row. Such output will be taken by EleExcel and copied to the output MExcel file without any change.

Referencing data attributes (fields) in EleExcel template files

Every cell in an EleExcel template file, which is to be filled with some data from the Elestore database, must contain a reference to an attribute of the data object class in Elestore, which is processed by this template. Please note that we are using the term “attribute” while the terms “data field” or “variable” are also frequently used. In most cases they can be used synonymously.

Please look into one of the sample templates and it will become immediately clear for every user of MExcel how these references into the database work.

EleExcel references are data to MExcel

Essentially, these EleExcel references are very similar to the MExcel VLOOKUP statement and to references to cells in other spreadsheets. Only their syntax is slightly different. Another difference is in the fact that they are interpreted by the EleExcel software, which drives MExcel, rather than by the MExcel software itself.

It is important to understand that these EleExcel references are nothing but data from the point of view of the MExcel software. As a consequence, such references are not changed when columns are inserted into or deleted from a template sheet.

Identifying an attribute in an EleExcel reference

The attribute, which is referenced in a cell, is identified through the attribute name as used in the Elestore database. Please look into the Elestore user manual to understand how attributes are named in Elestore. We use a very simple but concise and obvious naming convention.

Attributes in EleExcel templates can be referenced directly or indirectly:

- In the case of a direct reference to an attribute the attribute name is stored directly inside the EleExcel cell, which is supposed to be filled with the value of this attribute.
- In the case of an indirect reference to an attribute the cell contains a link to some other cell, which is typically located outside the print area and which contains the name of the attribute. This indirect referencing of attributes allows a simpler and quicker change of the content of cells in your EleExcel template file.

Referencing data objects (records) in EleExcel template files

Data objects are referenced only inside non-repeatable EleExcel templates.

In non-repeatable templates the identification of the data objects, whose data is to be output, can be stored either directly inside the respective MExcel cells or via a reference in separate column, which is typically located outside the printable area.

We strongly recommend you the second alternative and to place the keys to identify data objects in separate columns outside the printable area. This is much easier to maintain and you can also use copy/paste to fill in the needed data.

In repeatable templates the identification of data objects is supplied by EleExcel and therefore not contained in the template file itself.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Row type identifiers in column A of EleExxel templates

Overview on row types in column A in EleExxel templates

Column A of an EleExxel template sheet is reserved to contain a row identifier for every row, which tells EleExxel what to do with and how to handle this particular row. Column A can either be made invisible automatically or even removed from the EleExxel output file. Both are done by EleExxel depending on your chosen setting in the EleExxel project.

Important:

Column A of every row in your EleExxel template sheet must contain one of the row type identifiers.
Use the sample template files to better understand how these row identifiers are used.

List of row type identifiers for columns A

The following tags are permitted values in column A and are used for identifying the purpose of the row. Note that any other tag not listed will be discarded and such rows will be handled as though they contain the row type NOP.

Title rows in EleExxel - row type 'title' (TitleX)

This tag in column A identifies a title row, which is either placed only once at the top of an output EleExxel sheet or which is repeated after every number of data object instances or data rows as defined in the project settings.

The X identifies the title row number.

Title rows in an EleExxel template spreadsheet are taken by EleExxel and exported only once at the top of the output spreadsheet so that they indicate the purpose of the data rows and the meaning of the data in the data rows.

It is up to the user to decide whether or not these title rows are to be printed on the top of every page, which is formatted and printed by MSeExcel. This is a feature provided by MSeExcel, which is useful to repeat title rows on every new page.

Of course, title rows make especially sense in repeatable projects.

Data rows in EleExxel - row type 'data' (DataX)

This tag in column A identifies a row in a repeatable EleExxel template sheet. Rows with this tag are duplicated into the output EleExxel sheet and filled with data as defined by the cells in the other columns.

The cells in data rows in an EleExxel template spreadsheet are used, as the name says, as templates, which are filled with data from the Elestore database and which are then copied into the output MSeExcel spreadsheet. The cells in data rows in the template sheet define the data fields, which are exported, as well as the formatting of the cells in the data rows.

In order to achieve this, the cells of a data row contain VariableTemplates, which indicate the name of an attribute (data field) whose value is to be filled into this particular cell. Please refer to the chapter about VariableTemplates for further explanation.

The definition, formatting and usage of data rows is discussed in detail further below. They are the most essential components of an EleExxel template spreadsheet.

Row type 'cell' for cell references

xxxxxxxxxxxxxxxxxx

Row type 'col' for content references

xxxxxxxxxxxxxxxxxx

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Row type ‘cont’ for content (ContX)

This tag in column A identifies a row in a non-repeatable EleExxel template sheet. Rows with this tag are filled with the data as defined by the cells in the other columns. These rows are not duplicated, but for every row in the EleExxel template sheet an identical row filled with data from Elestore is placed into the EleExxel output sheet.

Row type ‘NOP’ or empty

This tag in column A identifies NOP or simply an empty cell in the first column of the EleExxel template sheet indicate that this row contains no functionality or data and it will therefore not be filled with anything. It will be output by EleExxel as an empty row.

Such rows should not contain any data or references, because they would be disregarded. NOP rows are output as empty rows into the output EleExxel sheet. (NOP stands for “no operation”; it’s an old and frequently used IT acronym)

Row type ‘REM’ for remarks

This tag in column A identifies a row in any type of EleExxel template sheet, which contains remarks by the user and which is neither to be output nor to be duplicated. Rows with this tag are skipped when reading the EleExxel template and are treated as though they would not exist in the EleExxel template sheet. This allows you to add your comments into an EleExxel template sheet, which will be disregarded by EleExxel as though there would not exist.

Row type ‘TITLE’ for title rows

xxxxxxxxxxxxxxxxxx

VariableTemplates in an EleExxel template file

Definition of VariableTemplates

A VariableTemplate is a placeholder stored in a cell of an EleExxel template spreadsheet. This placeholder indicates to the EleExxel software the value of which variable is to be inserted into this particular spreadsheet cell. During the generation process, the EleExxel spreadsheet generator replaces this VariableTemplate with the value of the variable of the processed data object from the database.

VariableTemplates for static attributes

VariableTemplates for static attributes consist simply of the name of the attribute (data field). The term “static attribute” is used to distinguish the “hard-coded” attributes as defined by the software from the dynamic attributes, which can be defined by the user.

With “name of an attribute” we mean the internal name of an attribute, which is always unique. This is not the visible field name (label) that is contained in the forms. This visible field name is what we call the field label. For details please refer to the Elestore user manual.

The visible field label is not unique and therefore does not provide a definite identification of a single datum in the Elestore database. Remember that this visible field label is also language dependent so that it will have a different identification depending on your current user interface language. Second, this visible field label can be changed by you by simply overwriting the text key translation file. Please refer to the chapter about the user interface languages and text keys in the Elestore user manual for further details.

Therefore, you have to add the internal name of an attribute (field name) to a cell in the EleExxel template spreadsheet. This VariableTemplate will then be replaced by the EleExxel generator with the value of this particular attribute from the currently processed data object. The template is exchanged with the real value in the database.

Please look at the published sample EleExxel template files where you find, among others, the variable names art00020PrimNrExt and art05011TextShort1 indicating, that the first one is the VariableTemplate for the external article number and the second one is the VariableTemplate for the first article short text. This is just an example. You can use any variable you like.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

VariableTemplates for dynamic attributes

Dynamic attributes are frequently used by Elestore and they differ from static attributes in so far as every data object can own a different set of dynamic attributes. For example, the eClass product classification system delivers attribute definitions for most product groups. So does Etim. If you link an article object to one of these classification levels the article object will automatically receive all the attribute definitions provided by eClass for this particular classification level.

A practical example: Of course, a screwdriver has a completely different set of technical attributes as a drilling machine and this is exactly what dynamic attributes exist for, no matter whether they are user-defined or defined by eClass.

Therefore, the way how VariableTemplates are used for static attributes would not be suitable for dynamic attributes.

We have therefore developed a form of relative VariableTemplates, which refer to the dynamic attributes of any article object in a relative way. As you can see from the published sample EleExxel spreadsheet file the format of these dynamic VariableTemplates is as follows: #DynValue01, #DynValue02, #DynValue03 and so forth up to #DynValue05 for the fifth dynamic attribute of this sample article object.

As always, these references to variables are enclosed in <...> brackets like this: <#DynValue01>.

Assuming, that a particular article object has five different dynamic attributes then #DynValue01 would be the VariableTemplate representing the first of these dynamic attributes, the VariableTemplate #DynValue02 would represent the second of these dynamic attributes and so forth.

The order of dynamic attributes is typically provided by Elestore. In the case of eClass or Etim the order of dynamic attributes is even provided by these classification systems.

By defining relative VariableTemplates you can provide as many data cells to be published for every article object as you like – up to the unrealistic maximum of 99 variables per data object. Of course, this does provide more than enough cells to cover the maximum number of dynamic attributes available for any published article object.

For those article objects with less dynamic attributes than you have prepared cells in your EleExxel template spreadsheet the remaining unused cells are not needed and these VariableTemplates and their cells will be skipped automatically by the EleExxel spreadsheet generator. Thus, only the needed space will be used so that you can specify enough dynamic VariableTemplates to suit the maximum number needed.

[Where to find the attribute names](#)

You find the attribute names in the database handbook and also in the Elestore attribute administration browser. Another place where they are presented is in the import and in the export managers.

VariableTemplates for names of dynamic attributes

In the case of dynamic attributes it is often desirable to also publish the field labels (visible names) of these attributes that are the external names in the currently selected user interface language, the same field labels that you see on forms.

This makes much sense, because in the case of dynamic attributes the title rows cannot indicate the name and purpose of the published data, because dynamic attributes are, as the name implies, dynamic and that means that they are dynamically allocated to articles so that not every article object owns the same set of dynamic attributes. Therefore, the number and the selection of dynamic attributes will typically vary from one article object to another.

This is why the values of dynamic attributes are typically published as pairs of field label (visible name) and value.

Normally, first the field label is published and on the right-hand side of the field label the attribute's value is published. Of course, it would also be possible to arrange this in a way that the label is on top of the value.

We have developed a form of template referring to the external names of dynamic attributes. As you can see from the published sample EleExxel spreadsheet file the format of these templates for the names of dynamic variables is as follows: #DynName01, #DynName02, #DynName03 and so forth.

As always, these references to names of variables are enclosed in <...> brackets like this: <#DynName01>.

Please note that the external attribute name (label), which replaces these templates, depends on the currently selected user interface language. This enables you to produce EleExxel spreadsheet files not only with language dependent variable content but also with the language dependent names (labels) of the exported attributes.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Defining the exported attributes (data fields)

The data fields of the article objects, which are to be exported by EleExxel, are defined in the template files by VariableTemplates. In this respect EleExxel’s way of exporting data from Elestore is very much different from the normal export manager of Elestore.

Direct and indirect attribute references

A cell in an EleExxel template sheet can either contain a direct reference to an attribute or an indirect reference.

Direct attribute references in cells

In the case of direct references the name of the referenced attribute of the data object is contained directly inside the cell, which is supposed to be filled with the value of this attribute during the generation process.

This is a sample referring directly to an attribute:

#art00020PrimExtNr

This is explained in more detail further below.

Indirect attribute references in cells

Indirect references are those where the cell of the template sheet, which is supposed to be filled with the value of an attribute during the generation process, is referencing another cell in the same template file, which then actually contains the name of the attribute.

Let’s look at an example: Cell B15 contains an indirect reference in so far as it refers to cell H15, which contains the name of the referenced attribute. Indirect references offer advantages and greater flexibility.

This is a sample referring indirectly to an attribute:

<!Col:G>

In this case the name of the attribute (field name) must be contained in column G of the same row.

This is explained in more detail further below.

Syntax of direct attribute references in EleExxel templates

Attributes are referenced by filling a cell with the name of the attribute preceded by the hash sign “#” (without quotes). These references are called “direct references”, because the name of the referenced attribute is contained directly in the cell of the template file, which is to be filled with the value of this attribute during the generation process. Further below in this chapter you will see that there are also indirect references available.

The syntax of direct references must be used as shown in this example: #art00020PrimNrExt

In this case the actual name of the attribute is "art00020PrimNrExt" (note that the quotes are not part of the name of the attribute). The hash sign (“#”) must always be added as the first character of the reference to an attribute.

This means that the whole attribute reference is always composed of the hash sign as the first character plus the name of the attribute, which must be correctly spelled and without any other character between the hash and the attribute name. Here is an example with three references, which we have embedded into a simple table to simulate the MSExcel view:

Data1	#art00020PrimExtNr	#art05011TextShort1	#art07511Length
-------	--------------------	---------------------	-----------------

- The first cell symbolises the row definition column A, which defines for EleExxel the purpose of this row.
- The second cell contains a reference to the primary external article number, which can be used as the key to identify an article object.
- The third cell contains a reference to the field article text 1 of the article object. This is a string field.
- The fourth cell of this example contains a reference to a numeric field, which contains the value for the length.

Please note that a different syntax must be used for indirect references (see further below).

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

In all of these cases the cell holding the attribute reference is filled by EleExxel with the value of this attribute from the currently processing article data object during the generation process. The cell indicates which attribute value to use.

Syntax of indirect attribute references in EleExxel templates

Indirect references refer to a cell, which is normally located outside the printable area. In these cases the template sheet as well as the final output sheet is de facto separated into a printable and a non-printable area. The non-printable area contains the real attribute names.

It would be wise to use several columns for these attribute names, simply because the values in a non-repeatable template can then be exchanged much simpler and quicker (for a repeatable template this makes no real difference).

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Here is an example with one indirect reference, which we have embedded into a simple table to simulate M\$Excel:

A	B	C	D	E	F	G
Data1	<!Col:G>	abc	def	Blah blah	#art05011TextShort1	#art00020PrimExtNr

- The cell in column B contains an indirect reference to cell G of the same row.
- The cell in Column G contains the name of an attribute and this must be preceded by the hash sign (“#”).
- The cell in Column F contains a direct reference: This shows us that direct and indirect references can be mixed in one template sheet.

Advantages of indirect references

Indirect references have great advantages primarily in the case of non-repeatable templates. Typically, one would equip the printable area of such a non-repeatable template sheet with indirect references, which all point to different columns outside the printable area. In such cases one would probably use one column for every type of attribute. In other words, one would place the real attribute names each into a different column and point to this column from the cell in the printable area, which is supposed to take up this value.

The advantage of this design of one template file is in the fact that one can very easily exchange the attribute names if they are all contained inside one column. This will often be safer than exchanging attribute names by an automatic search and replace. However, all of this is just a proposal for designing a template sheet and it is by no means a mandatory form of creating EleExel template files.

Another reason: You can much easier duplicate layouts by copy/pasting them to create additional pages. In this case you just copy/past the layout part and then add the references to the article objects in the columns in the non-printable area. This is much quicker than replacing the article references inside the layout.

Defining the exported data objects (instances, records)

Overview on defining the exported data objects

It depends on the type of template how and where the data objects are defined, which are to be exported:

- In the case of a repeatable template the data objects are selected by the user in one of different ways in the project form by performing some search, by selecting specific objects or one of the other ways. Please refer to the chapter about the EleExel project settings for a full description.

In some versions of Elestore the exported data objects can also be defined by so-called StoredSelections. They are permanently stored, user-defined selections (data sequences) of any number of data objects in any specific order.

In any of these cases a collection of article objects as defined by the user is used to provide the EleExel generator with the data objects for the output.

- In the case of non-repeatable templates the data objects to be exported are referenced directly in the non-repeatable template sheet. These data objects are defined by one of their article numbers (primary keys).
- In the case of ElePricelist using EleExel projects both variants can be freely mixed with one another. For further details please refer to the documentation and user manual of ElePricelist.

Referencing data objects (records) in repeatable and non-repeatable EleExel template files

The use of references to data objects depends on the type of the EleExel template file. In this respect we have to clearly distinguish between repeatable and non-repeatable template files.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

[References to data objects in repeatable EleExel templates](#)

First of all, we have to recognise that data objects are never referenced in a repeatable EleExel template file. This is the case, because it is a fundamental feature of repeatable EleExel template files that the data objects, whose data is to be filled into these templates, are determined by the user prior to starting the EleExel generation process. The data objects for repeatable template files are searched, selected or defined by the stored selection. In other words, there are never any references to data objects in repeatable templates, because they are added automatically by EleExel.

[References to data objects in non-repeatable EleExel templates](#)

Non-repeatable EleExel template files must contain references to data objects to work properly. Please read the following chapter about the syntax of references to data objects where the syntax of these references is explained.

We recommend you to store these references in separate cells outside the print area and to reference these cells, which contain the real references to the data objects, from within the cells in the print area. These allow us a much simpler and much more flexible way of changing the data objects used inside a non-repeatable template file.

StoredSelections for exporting data objects

Additionally, in some versions of Elestore so-called StoredSelections of article objects can be used and stored for re-use. A StoredSelection consists of an ordered number of references to article objects. A StoredSelection is either created automatically by searching for some keywords and/or manually entered. In any case it can be edited and ordered by the user.

A StoredSelection constitutes a precisely defined number of data objects, which is stored as an entity so that it can be easily and quickly re-used without any need to search or define again, which data objects are part of this selection.

Syntax of data object references in EleExel templates

Non-repeatable templates must also contain links to the data objects, whose values are supposed to be filled into the template sheet. Therefore, non-repeatable templates must contain references to attributes as well as references to data objects. Both references are included in one link.

[Example for an indirect link to data object and attribute in a non-repeatable template](#)

Here is an example:

A	B	C	D	E	F	G
Cont1	<!Col:F;#art00020PrimExtNr>	abc	def	Blah blah	EME X15-320	#art00020PrimExtNr
Cont2	<!Col:F;!Col:G>	afr	Hui	Blah blu	EME X15-321	#art00020PrimExtNr

- The cell in column B contains a combined reference to a data object and to an attribute. The link is explained below.
- The cell in column F contains the primary key of the data object, which is to be used to fill the cell in column B.
- The cell in column G contains the name of an attribute, which must be preceded by the hash sign (“#”).

[Syntax of the combined reference to data object and attribute](#)

The reference consists of several components, which are explained from left to right:

- The reference must start with the characters “<!” (as always without the quotes).
- The next part consists of the characters “Col:” (again without the quotes), which indicates that this is the pointer to another column in the same row.
- Then follows the name of the column, where EleExel is to find the key for the data object. In the above example this is the “F”.
- Next part is the delimiter character “;” (semicolon), which separates the reference to the object in the left part from the reference to the attribute.
- Next part is a normal reference to an attribute with the hash sign (“#”) as shown above in the first row or it’s an indirect reference, i.e. a link to another column as shown above in the second row.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

- To close the reference there must be the “>” at the end.

For further examples please refer to the sample template files, which are shipped with EleExel.

Special handling of dynamic attributes in EleExel

Typically, article objects stored in Elestore contain dynamic attributes (data fields). Some of these dynamic attributes are defined by the classification systems supported by Elestore like eCl@ss, Etim or proficl@ss and they are automatically added to an article object as soon as an article is linked to a grouping level equipped with dynamic attributes. Please refer to the general description of dynamic attributes in Elestore manual for further explanations.

Due to the very nature of dynamic attributes the existence and the number of attributes vary substantially from one data object to another. Some article objects have four, others ten or even more and some have no dynamic attributes at all. And, of course, the type of attributes assigned to article objects varies also from one data object to another.

It is therefore hardly possible to predefine a format or to exactly define the number of spreadsheet cells to visualise the dynamic attributes of article objects. On the other side, exporting and printing dynamic attributes is essential to precisely characterise a specific data object (article, product).

This is why we have designed a specific form of exporting the dynamic attributes of data objects to an MSExcel spreadsheet. They are called “VariableTemplates”. In the case of dynamic attributes you don’t specify the exact attribute names but instead you fill cells with a placeholder referring to the sequential number of every dynamic attribute for each particular data object.

The EleExel spreadsheet generator will then take all of the dynamic attributes of every processed article object and fill them into the cells marked with these dynamic VariableTemplates. The order of dynamic attributes is defined in Elestore or even by the grouping systems providing dynamic attributes.

Example: Special handling of dynamic attributes in EleExel

This is best explained by a little example: We want all the dynamic attributes of articles to be contained in the MSExcel sheet but the number of dynamic attributes per article varies substantially. In this case we have to take these precautions:

First, we select all the attributes, which are to be published. This makes all attributes available to EleExel except those, which were not activated. This is done in the export project in the export manager browser.

Second, we prepare the MSExcel template file with sufficient cells to present even the maximum number of dynamic attributes including their attributes names (labels). In the case of dynamic attributes it is typically desirable to also publish the attribute names, because they cannot be properly defined by title rows as is the case for static attributes.

The following excerpt of an EleExel template file shows how dynamic attributes are placed into cells using VariableTemplates:

The term VariableTemplate is explained in the special chapter on VariableTemplates. The VariableTemplates used in this case are special in so far as none of them directly refers to a specific attribute (data field) of the article object but instead they refer to one of the dynamic attributes.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Here is a symbolic sample excerpt from an EleExcel template spreadsheet file (some rows are omitted here):

A	B	C	D	E
Data04	#DynName01	#DynValue01	#DynName02	#DynValue02
Data05	#DynName03	#DynValue03	#DynName04	#DynValue04

Column A indicates the purpose of these template rows; in this case these rows are to contain data fields. Columns B and D hold templates for the names (labels) of dynamic attributes. Columns C and E hold templates for the values of dynamic attributes.

After being filled with data the output rows created from these template rows might look like this (column A has been made invisible, because it's not wanted in the output):

B	C	D	E
Height	100 mm	Width	150 mm
Colour	Blue	Weight	1200 g

Special handling of text data fields in EleExcel

General remarks on long floating texts in MExcel spreadsheet cells

One of the most important specialities of Elestore is its capability to store multiple multi-lingual floating texts in RTF or MSWord format or as MExcel files. In this respect MExcel files are processed almost in the same way as MSWord files. However, let's concentrate on RTF and MSWord floating texts, because it neither makes sense to try to integrate MExcel spreadsheets into a cell of another MExcel spreadsheet nor is it technically possible. However, MExcel sheets stored in the Elestore database or generated by EleExcel can be used by ElePricelist and embedded into MSWord documents.

EleExcel is well capable of generating spreadsheet cells with the content of RTF or MSWord floating text variables stored as part of article objects in the Elestore database. However, the integration of long floating texts into MExcel spreadsheet cells will soon reach a natural limit inherent to the nature of the MExcel spreadsheet software.

The more recent versions of MExcel are capable of storing long floating texts in a spreadsheet cell. Still, MExcel was not designed to handle long floating texts. This is why one has to live with certain limitations such as:

- Individual formatting of floating text inside a spreadsheet cell is very problematic in practice.
- Most of the formatting of an MSWord text is lost when it is imported into a spreadsheet cell of MExcel unless it is integrated as an embedded object.
- The handling is quite awkward when editing long floating texts inside a spreadsheet cell.
- Difficulties arise in formatting layout and pages when cells contain long floating texts.

We therefore recommend you to carefully decide about the size, structure and formatting of your long floating texts, which are supposed to be integrated into the cells of an output EleExcel spreadsheet.

In practice, it is quite well possible to integrate floating texts of a few lines and up to a few hundred characters size into a spreadsheet cell. But the longer the texts are the more difficult the handling will become. From a certain moment on it is probably much more practical to use MSWord as the output format and ElePricelist as the generator software.

Please look at ElePricelist, which offers much better formatting capabilities on long floating texts by using MSWord. If you have many long and formatted floating texts in your output it might be better to use ElePricelist and to output the result to MSWord.

Formatting of data rows in EleExcel template files

Data type formatting and font related formatting

When it comes to formatting a cell in the template spreadsheet we have to distinguish between assigning the spreadsheet cell a specific data type such as a day's date or a currency and the font related formatting. These are two separate issues.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Data type formatting of EleExxel template cells

MSEExcel distinguishes between different data types. A cell can be assigned to one of these data types. For further details please refer to MSEExcel menu item: Format/cells/number and look at the selection list for category. Please also refer to the MSEExcel help and documentation. Of course, our support will be pleased to assist you and they are also ready to develop your individual template files for you.

As far as the formatting of the cells in the EleExxel template spreadsheet is concerned there are two possibilities:

You assign the template cells to the data type corresponding to the data type of the attribute whose value this cell is supposed to contain and to visualise.

The other possibility is that you leave this task completely to EleExxel, which will assign the appropriate data type to generated cell corresponding to the basic data type of the attribute whose value is filled into a particular cell.

Note that the size of string content in cells is limited to 255 characters. Longer texts such as MSWord documents are included into a cell as text objects, which is different from a normal string in a cell.

Font related formatting of EleExxel template cells

A nice feature of MSEExcel is its capability to format every spreadsheet cell individually with specific fonts, font styles, font size and other characteristics. This was one of the main reasons why we choose MSEExcel as an output and print generator for the database content in Elestore.

You can and you should provide every cell of your EleExxel template spreadsheet with the appropriate font related formatting information. This will decide about the look and feel of the final output spreadsheet file.

The font related formatting is exclusively defined by the cells of the EleExxel template spreadsheet. Their font related formatting information is used and copied into the generated output spreadsheet by EleExxel.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleExcel repeatable templates

What means “repeatable template”?

When using “repeatable templates” EleExcel takes one MSeExcel template sheet and duplicates it as many times as there are article objects to export. Each of these repeated copies of the template sheet is filled with the data of one article object. As a consequence, each of these copies of the template file looks fairly identical in their layout although each of them contains different data.

This one repeatable template sheet therefore defines the layout and the formatting of all exported data objects although the number of generated rows can vary depending on the existence or absence of content data.

Repeatable templates contain only links to data fields but no references to data objects whose data is inserted, because the data objects (instance or records) are defined by searches, selections or externally by the operator.

This form of using EleExcel is the preferred one when a relatively large amount of data objects is to be generated fully automatically in a fairly uniform layout. Creating this one repeatable template is much simpler and quicker compared to the two other template types, which allow much more individual and varying layouts.

Any MSeExcel user can do. It’s easy, simple, quick, yet extremely powerful.

Structure of an EleExcel repeatable template

Sample EleExcel template spreadsheet file

This is the sample template file, which is discussed throughout this chapter. Your templates can look differently, of course.

	A	B	C	D	E	F	G
1	Title1	Article text 1			Article number	List price	
2	Title2	Article text 2			EAN		
3	Title3	Manufacturer			Manufacturer number	Price valid till	
4	Title4	Detailed product description		Picture			
5	Data1	#art05011TextShort1			#art00020PrimNrExt	#art07101Price01	
6	Data2	#art05012TextShort2			#BAA271001		
7	Data3	#art09015ManufacturerName			#art09020ManArtNr	#art00947PricingValidTill	
8	Data4	#art05050TextDescCommercial		#art05311Image01			
9	Data5	#DynName01	#DynValue01	#DynName02	#DynValue02		
10	Data6	#DynName03	#DynValue03	#DynName04	#DynValue04		
11							

Purpose of special column A in an EleExcel template

The column A is reserved as an identification column for the rows of an EleExcel template file. Column A will neither be filled with any data nor will it be part of the final printable MSeExcel file. Column A can be removed or made invisible in the final MSeExcel output file. You decide this in the EleExcel project settings.

Different types of rows in an EleExcel template

The type of a row in an EleExcel template spreadsheet is identified by the value in the column A. Currently we distinguish between those types of spreadsheet rows:

- Title rows contain column titles and are exported to indicate the purpose of the columns.
- Data rows are used to export user data.
- REM and NOP rows are ignored by the EleExcel generator.

For more details see the chapters “Discussion of a repeatable template for EleExcel” and the chapter “EleExcel templates – structure, format, data references”.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Discussion of a repeatable template for EleExcel

View on an EleExcel template spreadsheet file

This is a view on an EleExcel template file:

Title1	Article text 1			Article number	List price
Title2	Article text 2			EAN	
Title3	Manufacturer			Manufacturer number	Price valid till
Title4	Detailed product description		Picture		
Data1	art05011TextShort1			art00020PrimNrExt	art07101Price01
Data2	art05012TextShort2			BAA271001	
Data3	art09015ManufacturerName			art09020ManArtNr	art00947PricingVa
Data4	art05050TextDescCommercial		art05311Image01		
Data5	DynName01	DynValue01	DynName02	DynValue02	
Data6	DynName03	DynValue03	DynName04	DynValue04	

Explanation of the EleExcel template spreadsheet (sample)

This is a discussion of the structure of this sample EleExcel template file. You find a more detailed discussion on the purpose of the various template cells further below in this manual.

[EleExcel title rows in the sample template file](#)

The first four rows are marked as title1 through title4 in column A. This indicates that these rows will be output only once at the beginning of the generated MSeExcel sheet. These title rows can be printed on every subsequent page by using the special functionality of the MSeExcel software, which allows repeating title rows on every printed page.

The content of these title rows is output as it is. EleExcel makes no changes to the content of these rows. They can also be omitted.

[EleExcel data rows in the sample template file](#)

The next six rows are marked with data1 through data6 in column A. These rows are used by the EleExcel spreadsheet generator as templates to produce output rows of the same format, which are filled with the data values of the processed article data object. The content of the template cells are VariableTemplates, which refer to specific attributes (data fields).

[Static attributes in the sample template file](#)

The rows data1 through data4 refer to static attributes. The values in these template cells are VariableTemplates, which are explained further below.

[eCl@ss variable in the sample template file](#)

Cell E6 holds the VariableTemplate “#BAA27101”, which is taken from the eCl@ss attributes. This is explained in detail in the Elestore user manual in the chapter about the eCl@ss product classification system and its defined attributes. These VariableTemplates must also be preceded by a “#” as all others (without quotes).

[Dynamic attributes in the sample template file](#)

The rows data5 and data6 refer to dynamic attributes, which are in this case grouped in two logical columns, one in columns B/C and the other one in columns D/E.

The values in these template cells are dynamic VariableTemplates, which are explained further below.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleExel non-repeatable templates

What means “non-repeatable template”?

This type of EleExel template sheet allows much more individual and flexible layout and formatting structures. The fundamental difference to repeatable templates is in the fact that all of the pages already exist preformatted in the non-repeatable template file but yet without content, of course.

As the name says, non-repeatable templates are not duplicated but only filled with data by the EleExel generator. No automatic duplication of the templates is performed in this case. Instead, the non-repeatable template is filled with data from the database and the links to the data objects (instances, records) are already contained in the template file.

This non-repeatable type of EleExel template contains links to data fields as well as to data objects while the repeatable template type only contains references to data fields and the objects whose data is generated are defined by searches, selections or externally by the operator.

The automation offered by this non-repeatable template type is in the fact that the non-repeatable template sheet is filled fully automatically with varying data by EleExel. However, every page must already be contained in the template file, because no automatic duplication of repeated layouts takes place in this case.

The great advantage of the non-repeatable template type is in the very much higher flexibility of individually designing pages, which are then automatically filled by EleExel with up-to-date content from the database. Non-repeatable templates are ideally suitable for very individually formatted product lists or price lists where many or even every page can look very different from the rest.

Of course, some of this content can also consist of customer or publication specific content such as specific prices or different languages. One non-repeatable template can be used with different content, i.e. different prices and/or different languages. Even language dependent content that is not directly related to article objects can be imported from the Elestore database by referring to text objects whose content is generated by EleExel into the non-repeatable template.

Discussion of a non-repeatable template for EleExel

View on a non-repeatable EleExel template file

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Cont1	<!Col:S:#art05013TextShort3>													tShort2>				TL-1002				
2	Cont2	<!Col:S:#art05011TextShort1>																	TL-1002				
3	Cont3																		TL-1002				
4	Cont4		<!Col:S:#art05050TextDescCommercial>																TL-1002				
5	Cont5																		TL-1002				
6	Cont6																		TL-1002				
7	Cont7																		TL-1002				
8	Cont8																		TL-1002				
9	Cont9																						
10	Cont10	Rolladen-einschub-leiste																	TL-1002				
11	Cont11		weiß	hAttr>	hAttr>	hAttr>	hAttr>	hAttr>	hAttr>	hAttr>	hAttr>	hAttr>	hAttr>						TL-1002-50	TL-1002-50	TL-1002-50	TL-1002-50	TL-1002-50
12	Cont12																						
13	Cont13																						
14	Cont14	Breite	RAM	rt7500	rt7500	rt7500	rt7500	rt7500	rt7500	rt7500	rt7500	rt7500	rt7500	rt7500	1Paar								
15	Cont15	Hölle	rt7503	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	Rollschienen			TL-1002-50	TL-1002-50	TL-1002-50	TL-1002-50	TL-1002-50
16	Cont16			<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:					TL-1002-50	TL-1002-50	TL-1002-50	TL-1002-50	TL-1002-50
17	Cont17		rt7503	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501				TL-1002-62	TL-1002-62	TL-1002-62	TL-1002-62	TL-1002-62
18	Cont18			<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:					TL-1002-62	TL-1002-62	TL-1002-62	TL-1002-62	TL-1002-62
19	Cont19		rt7503	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501				TL-1002-75	TL-1002-75	TL-1002-75	TL-1002-75	TL-1002-75
20	Cont20			<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:					TL-1002-75	TL-1002-75	TL-1002-75	TL-1002-75	TL-1002-75
21	Cont21		rt7503	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501				TL-1002-87	TL-1002-87	TL-1002-87	TL-1002-87	TL-1002-87
22	Cont22			<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:					TL-1002-87	TL-1002-87	TL-1002-87	TL-1002-87	TL-1002-87
23	Cont23		rt7503	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501				TL-1002-10	TL-1002-10	TL-1002-10	TL-1002-10	TL-1002-10
24	Cont24			<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:					TL-1002-10	TL-1002-10	TL-1002-10	TL-1002-10	TL-1002-10
25	Cont25		rt7503	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501				TL-1002-11	TL-1002-11	TL-1002-11	TL-1002-11	TL-1002-11
26	Cont26			<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:					TL-1002-11	TL-1002-11	TL-1002-11	TL-1002-11	TL-1002-11
27	Cont27		rt7503	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501				TL-1002-12	TL-1002-12	TL-1002-12	TL-1002-12	TL-1002-12
28	Cont28			<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:					TL-1002-12	TL-1002-12	TL-1002-12	TL-1002-12	TL-1002-12
29	Cont29		rt7503	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501				TL-1002-13	TL-1002-13	TL-1002-13	TL-1002-13	TL-1002-13
30	Cont30			<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:	<!Col:					TL-1002-13	TL-1002-13	TL-1002-13	TL-1002-13	TL-1002-13
31	Cont31		rt7503	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501	rt7501				TL-1002-15	TL-1002-15	TL-1002-15	TL-1002-15	TL-1002-15

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

[Explanation of a non-repeatable EleExxel template file \(sample\)](#)

This sample spreadsheet template is vertically divided into the printable area in the left half and the right half, which is not printed. This was done intentionally in order to make the editing of the article number references easier and this is the case if they are stored outside the printable area.

In this case we have produced about 300 of such pages and most pages were created by copy/pasting the printable part from one page to another so that we only had to exchange the article numbers. This is another reason why we have chosen to let most of the cells in the printable area refer to a cell outside the printable area where the article number is really stored.

You see this a little bit better from the enlarged excerpt of the printable area and for this picture we have modified the column width so that the full formula inside the cells becomes visible.

[Enlarged view of some template cells with fully visible cell content](#)

	A	B	C	D	E	F
10	Cont10	Rolladen-einschub-leiste				
11	Cont11		weiß	<!Col:S;#art75020AchAttr>	<!Col:T;#art75020AchAttr>	hAttr>
12	Cont12					
13	Cont13	Breite				
14	Cont14		RAM	<!Col:S;#art75005AchAttr>	<!Col:T;#art75005AchAttr>	rt7500f
15	Cont15		<!Col:V;#art75034AchAttr>	<!Col:S;#art75018AchAttr>	<!Col:T;#art75018AchAttr>	rt7501f
16	Cont16	Höhe		<!Col:S;#art75004AchAttr>	<!Col:T;#art75004AchAttr>	<!Col:
17	Cont17		<!Col:V;#art75034AchAttr>	<!Col:S;#art75018AchAttr>	<!Col:T;#art75018AchAttr>	rt7501f
18	Cont18			<!Col:S;#art75004AchAttr>	<!Col:T;#art75004AchAttr>	<!Col:

This view is from the sample template shown further above.

[Discussion of the content of a cell](#)

As you can see most of the cells contain a formula like the following:

```
<!Col:V;#art75034AchAttr>
```

The left part of this formula, which represents the identification of an article object, refers to a column outside the printable area (here to column V). The second part of the formula, which represents the attribute (data field) is contained directly inside the formula, because in all of these cases the same attribute is published (of course, except for the title rows etc). Of course, another reference to some other column would also be possible but it would not have made sense in this example.

By referring to some column outside the printable area it is much easier to add and later edit the article numbers that are to be published on a certain page. Another great advantage is in the fact that one can easily copy/paste most of the printable area to a new page and one only has to exchanges the article numbers (references) in the non printed area.

In the case of this catalogue there were a bit more differences between the various pages, especially in the title part and often also in a footers. These differences were the primary reason why we had to choose non-repeatable templates and why we could not use repeatable templates, which would have been created much quicker and easier, of course.

[Font definition inside the template](#)

As you can see very well especially from the rows 1 and 2 of this example the formatting can be freely applied to the template and this formatting is used by EleExxel accordingly in the generated output pages as you can see from the view of the final page further below.

Note: This manual is "work in progress" and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

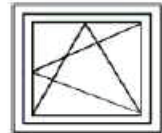
View on a the final page created from a non-repeatable EleExcel template file

This is the page, which was generated automatically by EleExcel from the above shown non-repeatable template:

Serie Achenbach Thermoline 1-flg. Kunststoff-Fenster DK

Typ 1002

Climaglas Ug 1.1 W/qmK nach DIN EN673
Basissicherheit, Fehlbedienungsicherung
Fenstergriff Rotoline weiß
ohne Wetterschenkel
Sohlbankprofil: 30mm 110.071Regenerat



Rolladen- einschub- leiste		weiß	5	7	8	9	11	12	13	15	16
Breite Höhe	RAM	500	625	750	875	1000	1125	1250	1375	1500	1 Paar Rolloschienen
	500	125 0,40	130 0,40	141 0,40	148 0,44	159 0,50	167 0,56	174 0,63	182 0,69	189 0,75	11
	625	130 0,40	136 0,40	147 0,47	154 0,55	165 0,63	173 0,70	181 0,78	195 0,86	204 0,94	14
	750	141 0,40	146 0,47	157 0,56	165 0,66	177 0,75	187 0,84	197 0,94	207 1,03	217 1,13	16
	875	148 0,44	153 0,55	164 0,66	174 0,77	188 0,88	199 0,98	210 1,09	221 1,20	232 1,31	19
	1000	164 0,50	169 0,63	181 0,75	194 0,88	209 1,00	220 1,13	231 1,25	243 1,38	254 1,50	22
	1125	171 0,58	176 0,70	191 0,84	204 0,98	220 1,13	232 1,27	244 1,41	256 1,55	268 1,69	24
	1250	178 0,63	184 0,78	200 0,94	214 1,09	230 1,25	243 1,41	255 1,56	269 1,72	281 1,88	27
	1375	185 0,69	194 0,86	210 1,03	225 1,20	242 1,38	255 1,55	268 1,72	282 1,89	295 2,06	30

The values inside the matrix are prices and they can be calculated individually by the MSeExcel template sheet. In this case EleExcel adds basic prices into some column outside the printed area and the visible prices are calculated from these basic prices by some user-defined formula. It is even possible to have EleExcel add calculation factors during the generation process and with Elecalc this whole price calculation process can be made even fully automatic.

Note that this was not wanted by this customer and therefore the above shown cells do not reference some calculation cells outside the printable area.

More templates and sample pages on our websites

Please look on our product websites for more template samples and for pages generated by these samples. Please refer to www.elestore.com or www.elestore.de in Deutsch and in early 2007 also to www.EleExcel.com (still in preparation).

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleExel projects and their settings

Overview and purpose of projects in EleExel

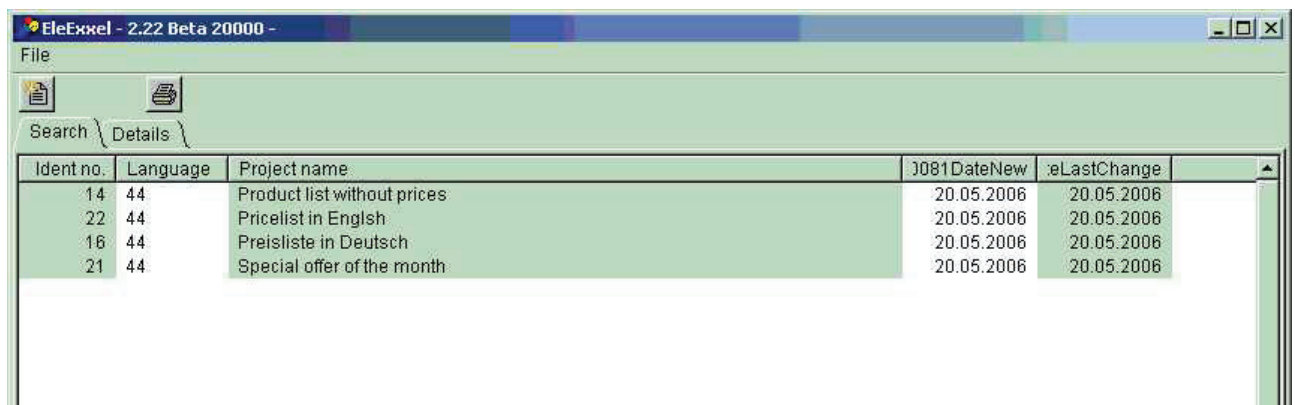
There are various settings, which decide about how the EleExel generator behaves when creating output and when adding values to the cells of your EleExel template spreadsheet. Some of these settings substantially influence the content and/or the look and feel of the generated spreadsheet output file.

All of the parameters and settings, which are to be used by EleExel for running a job, are entered and stored in an EleExel project for quick re-use.

We have implemented all settings and rules, which we thought to be of practical advantage for you as an EleExel user. However, EleExel is a relatively young product (first published in I. Q. 2007) and has therefore not yet been influenced too much from end-user's experiences. We invite you to please let us know your requirements for future releases of EleExel and we will be happy to implement such requirements by adding additional settings and rules in order to improve the usability of future EleExel releases. Please let us know your demands and requirements!

EleExel project table

This is a partial view of the project selection table of EleExel:



Ident.no.	Language	Project name	DateNew	LastChange
14	44	Product list without prices	20.05.2006	20.05.2006
22	44	Pricelist in English	20.05.2006	20.05.2006
16	44	Preisliste in Deutsch	20.05.2006	20.05.2006
21	44	Special offer of the month	20.05.2006	20.05.2006

You can select any EleExel project from this table for simple re-use. Click on one of the existing projects and then switch to the details form to re-use the stored project with or without modification of the settings.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

The EleExxel project details form

View of the EleExxel project details form

The screenshot shows the 'EleExxel - 2.3 Beta - Project browser EleExxel' window with the 'Details' tab selected. The form is organized into several sections:

- Settings:**
 - Project name: Project 450N
 - Name of template file: \Templates\EleExxel\Repeatable\Template-450N(rep).xls
 - Type of template: 10-Repeatable template (duplicate one template many times)
 - Export type: 200-Export all of a grouping level
 - Office product group: 9-450 N
 - Name of output file: \EleExxel\450N(rep).xls
 - Primary key to use: 10-Ident no. (art00010IdentNr)
 - Language: 44-English
- Handling of page titles:**
 - ☐ Remove title rows
 - ☒ Define title rows as header
- Handling of page breaks:**
 - Page break after...: 20-Insert page break after number of articles
 - Page break after number of...: 5
- Special rows and columns:**
 - Handling of column defining the row types: 20-Delete column
 - Column number defining the row types: 1
 - ☒ Remove comment rows
 - ☐ Remove empty rows
- Handling of formatting:**
 - ☐ Use formatting from MSWord
- Handling of auto-fit:**
 - ☐ Auto fit columns
 - ☒ Auto fit text objects
 - ☒ Auto fit pictures
 - ☐ Auto fit rows
 - ☒ Lock aspect ratio for texts
 - ☒ Lock aspect ratio for images

At the bottom right, there are three buttons: 'Cancel', 'Select articles', and 'Export'.

This is the EleExxel project settings form.

Menu and buttons in the EleExxel project settings form

[File menu in the EleExxel project settings form](#)

The file menu contains the usual menu items for adding, deleting, saving, or printing a project.

[Button Select articles in the EleExxel project settings form](#)

Pressing this button opens the form for selecting article objects. Please refer to the chapter below for a detailed discussion of the article selection form.

[Button Export in the EleExxel project settings form](#)

Pressing this button starts the processing of this project. If you have not yet selected the article objects the form for selecting article objects will be opened first.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

The settings of an EleExcel project

Project name

The field project name stores a text describing the purpose of this project. You should select some meaningful explanation of what this project does so that it will be easy for you or other users to select and re-use this project without having to look at all the settings.

Name of template file

This field stores the path and the name of the template file, which is to be used for this EleExcel project. Press the little button at the right end of the field and a file selection dialog form opens where you select the path and the file that is to be used as a template.

Please remember that EleExcel is not capable of detecting if the content and the structure of the selected template file corresponds to the other parameters of your project settings. This is your responsibility. You will get unexpected results if the structure and the content of the template file does not correspond to the project settings.

Type of template

Selecting the type of template is the most important of all settings. The type of template selected here must correspond to the content and formats of the template file, which you have selected in the field name of template file.

Please note that there is no possibility for EleExcel to check whether the type and the structure of a template file corresponds to the selected type of template.

Please read carefully the description of the different types of templates before selecting a type of template.

Trial version:

Users of the trial version are recommended to use one of the predefined projects where the correct type of template should be defined according to the sample template files delivered with the trial version.

If you change the type of template or the template file in a project you may get unexpected results if the template file does not correspond to the selected template type.

[Repeatable template \(duplicate one template many times\)](#)

This type of template file is used and repeated for every data object, which is to be exported. A copy of this template file is created by EleExcel and filled with data from the Elestore database and added to the output file. This is repeated as many times as there are selected data objects to be exported.

Repeatable templates are best suitable for consistently structured tabular lists of data objects.

In repeatable template files only the attributes (data fields) and no data objects are specified in the template file. The data objects are defined by the user selected data objects in the article selection form.

[Non-repeatable template \(fill all in one template\)](#)

This type of templates is used only once and it is not repeated. Non-repeatable templates are filled with the data from the data objects, which are referenced in the non-repeatable template file.

Non-repeatable templates are best suitable for varying layouts, which differ from one product to the other and within the various pages.

When selecting this template type the selection of article objects for output is disabled, because the data objects, which are to be included in the output, must be defined inside the template file.

In non-repeatable templates both the data objects (records) and their attributes (data fields) are specified inside the template file. Therefore, no data objects can be defined in the article selection form.

[Autonomous pages \(one template per page\)](#)

This type of template is used for creating single pages each of which contains the content of one data object. As for the selection of data objects autonomous page templates are handled similar to repeatable templates in that the data objects are selected by the user in the article selection form.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

For each of the selected objects one page is created from the selected template file. In most cases this type of template is used to create HTML files.

If you activate the checkbox “Templates from objects” EleExxel uses the template files stored in the processed data objects for creating pages. In cases where no template file is stored inside the data objects the default template file specified in this project is taken instead. This allows you to define individual template files for every single data object.

Depending on the output media the definition of “page” does not imply any specific format, because in the case of HTML page can be larger than a standard paper page.

Output of single pages to HTML
the output generated from single page templates to HTML is not available in all versions of EleExxel.
This feature is one of the primary features of EleProductSite.

This templates type is not available in all versions of EleExxel.

With inheritance

Provided that your version of Elestore supports data inheritance you can select or deselect data inheritance for your current project with this checkbox.

If you deselect data inheritance and your data has been stored with active inheritance the output data will most likely not correspond to what you have expected, because in these cases only the private data of child objects will be exported. On the other hand, there are occasions where this is exactly what one wants, for example when you want to control the private values of child objects.

For a more detailed discussion of data inheritance please refer to the corresponding chapter in the Elestore user documentation. Please note also that some versions of EleExxel do not support data inheritance and that in these cases you cannot activate this checkbox.

Export type

This setting defines for all repeatable templates the scope and number of data objects, which are to be exported. The data objects are searched and selected in the article selection form. In some versions of Elestore and EleExxel StoredSelections of data objects can also be used.

[Export selected objects \(110\)](#)

This setting exports only those objects, which were individually selected by the user in the article selection form. Selecting objects in this sense means that single rows or the range of rows have been selected in the search results table.

[Export single block of objects \(120\)](#)

This setting exports the current block of data objects from the search result form. Please refer to the documentation of Elestore for a more detailed explanation of what a block of objects means in this sense.

[Export all for search condition \(130\)](#)

This setting exports the entire search result as defined by the search condition in the article search form.

[Export of a grouping level \(200\)](#)

Selecting this export type activates the selection field “Office product group” in the project settings. With this export type EleExxel will traverse all article objects, which are linked to the selected office product grouping level and its sub-levels and EleExxel will use these article objects during the generation process.

With this export type you can use the office grouping system for defining the scope and the sequence of article objects, which are exported by EleExxel.

Office product group

This field is only available if you have selected in the field type of template the type 200, that is export all of a grouping level. In all other cases this field is disabled.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

This field allows you to select a grouping level from the office grouping system (please note that there are typically several different grouping systems available in EleExxel). By selecting a specific grouping level you tell EleExxel, which article objects to process in which sequence, which is both defined by the grouping level and its article objects.

For further details please refer to the chapter about the office grouping system.

Name of output file

Here you can add a file name for the MSeExcel output file, which is created by the EleExxel output procedure. By default, EleExxel will name the newly created and stored MSeExcel file using the current date and time in the sortable format EleExxel_YYYY.MM.DD_HH.MM.SS.xls.

If you have selected an export type that produces single pages then EleExxel will use the values of the primary key of the processed data objects to name the created files.

Overwrite existing output file

If you activate this checkbox existing files with the same name as the created output file(s) will be overwritten without asking. Please note that the output of EleExxel is always written to the standard output directory as defined in the general system settings.

Primary key to use

If you have chosen a non-repeatable template type the references to data objects are stored inside the template file. As references one of the key fields of the article object must be used.

There are two different primary keys, which can be used to identify article objects. In this selection field you decide which of those primary keys to use. This must correspond to the references to data objects in the selected template file, because otherwise Elestore will not be able to identify and load the data objects from the database.

Ident no. (art00010IdentNr)

This is the internal primary article number (ident number), which uniquely identifies every data object in the Elestore database. The ident number is purely numeric and it is always unique within one class of objects (table) in your database. Please refer to the Elestore user documentation for a detailed discussion of the ident number.

External art no. (art00020PrimNrExt)

This is the primary external and alphanumeric article number, which can also be used under certain circumstances to uniquely identify article objects. However, it depends on your settings in the general user settings if this type of article number is really unique within your system.

Save also as HTML

If this checkbox is activated EleExxel will automatically save all generated output files also in HTML format additionally to the MSeExcel format (*.xls).

Note that the HTML file can look different from the MSeExcel file as the HTML reproduction is often not absolutely identical to the original, which is mostly due to the conditions and rules applying to HTML and Internet browsers. Such differences are not a fault of our software but inherent to the rules and concepts of HTML and Internet browsers.

Split HTML pages

If this checkbox is activated EleExxel will automatically split the exported HTML pages of a repeatable project into single pages. This allows creating single HTML page files from the content of a repeatable MSeExcel output sheet.

Language

Elestore and EleExxel support multi-lingual content, of course, because the world is and will always be multi-lingual.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

In this selection field you decide about the language, which is to be used for all language dependent data fields such as strings and text fields. This setting allows you to use one template file for any number of languages provided that all the language dependent content comes from the Elestore database.

Tips and tricks:

In most cases some text will be directly included in template files, which is not imported from data objects. This is often the case for titles or footnotes. Such texts make a template file language dependent.

However, you can easily avoid this by creating some fake article objects, which only exist to act as holders for such language dependent texts. You then include references to these special fake objects in order to make your template files fully language independent.

The settings of an EleExcel project - handling of page titles

Remove title rows

If you activate this checkbox EleExcel will remove the title rows from the generated spreadsheet file. This is for cases where you don't want title rows to be published although they are contained in the template files.

Define title rows as header

If you activate this checkbox EleExcel will export the title rows only once to the first page and it will define the title cells in the output MSeExcel file as headers so that they are automatically duplicated on every printed page by MSeExcel.

Page breaks after article or rows

This field is only available if you have selected to use a repeatable template. It would make no sense for non-repeatable templates, because page brakes should be defined within non-repeatable templates themselves.

This setting allows you to tell EleExcel to automatically add page breaks. Either you decide to add a page break after a given number of objects or after a given number of rows exported to MSeExcel.

No page breaks to be inserted

If you select this setting EleExcel will not include any page breaks into the output file. You can still let MSeExcel calculate the page breaks itself.

Insert page break after number of articles

If you select this setting EleExcel will include a page break into the output file after every number of articles as specified in the field “Add page breaks after X articles”.

Insert page break after number of rows

If you select this setting EleExcel will include a page break into the output file after every number of rows as specified in the field “Add page breaks after X rows”.

Note that title rows are only added by EleExcel to the first page. If you want the title rows to be repeated on every page you must activate this feature in MSeExcel.

Add page breaks after X articles or rows

This setting allows you to tell EleExcel to add a page break after any number of exported article objects or rows. Either you decide to add a page break after a given number of rows or after a given number of objects exported to MSeExcel.

The settings of an EleExcel project - handling of rows and columns

Handling of column defining the row types

Make column A in the final EleExcel output file invisible

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

If you activate this setting the column A will be made invisible in the final EleExcel output spreadsheet file. Activating this setting conflicts with an activated setting for removing column A from the final MSeExcel file. You have to decide between either of both.

Hide column

If you select this setting the row definition column in the output file will be hidden.

Delete column

If you select this setting the row definition column will be deleted from the output file.

Show column

If you select this setting the row definition column in the output file will be visible (normal, not hidden).

Column number defining the row types

By default, column A (= column number 1) contains the row types. This field allows you to define some other column than A to hold the row types. Enter 2 for column B or 3 for column C etc.

Remove comment rows

If you activate this checkbox all comment (remark) rows are removed from the generated output file. Comment rows are those that contain the tag “REM” in the row definition column.

Remove empty rows

If you activate this checkbox all empty rows are removed from the generated output file. This is especially meant to remove rows reserved for attributes, which do not contain values and are therefore empty in those cases.

Use formatting from MSWord

This checkbox is related to cases where the content of an MSeExcel cell consists of an MSWord document. If you activate this checkbox the formatting of this embedded MSWord text will be kept. If this checkbox is unchecked the formatting of the MSeExcel cell will be used and this will overwrite the formatting of the enclosed MSWord text.

The settings of an EleExcel project - handling of auto-fit

Auto-fit column

If you activate this checkbox EleExcel will tell MSeExcel to apply the auto-fit function on all columns.

Auto-fit rows

If you activate this checkbox EleExcel will tell MSeExcel to apply the auto-fit function on all rows. MSeExcel then automatically adjusts the height of rows to the maximum height needed to fully visualise the content of cells. This is especially valuable for cases where long texts are contained in an MSeExcel cell, which need more space than is available from the standard height of the row to be fully shown.

This does work if the row would become smaller than the default row height of MSeExcel.

Auto-fit text objects

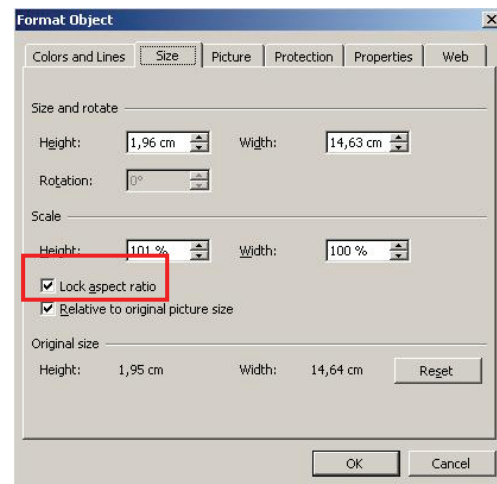
If you activate this checkbox EleExcel will tell MSeExcel to apply the auto-fit function on all cells holding texts or embedded text objects.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Lock object aspect ratio for text cells

If you activate this checkbox EleExxel will tell MSeExcel to keep the aspect ratio when applying the auto-fit function on all cells holding text objects.

On the right you see a screenshot of the corresponding settings form in MSeExcel.



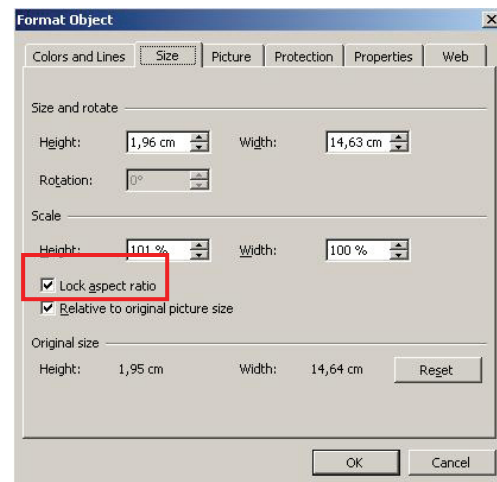
Auto-fit pictures

If you activate this checkbox EleExxel will tell MSeExcel to apply the auto-fit function on to all cells holding pictures.

Lock aspect ratio for picture cells

If you activate this checkbox EleExxel will tell MSeExcel to lock (= keep) the aspect ratio when applying the auto-fit function on all cells holding pictures.

On the right you see a screenshot of the corresponding settings form in MSeExcel.



Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleExxel tips and tricks

Use of the print area

Please remember that you can define a print area in MSeExcel. This is a very good solution for omitting certain parts of automatically generated EleExxel spreadsheets from printing. Like this you can add columns outside the designated print area where you can store calculation formulas, which are to be used on the generated data.

A practical example:

You use columns A through E for holding the output data that is automatically generated by EleExxel. However, column D contains some prices, which are supposed to be calculated from database data copied into some cells by EleExxel and by formulas, which are stored elsewhere in the EleExxel template sheet.

In this example EleExxel would copy some base price from the database into the cells of column E. The final price calculated by some formula is to be contained in column D. You place some formula into the cells of column D, which refer to the cells of column E where EleExxel will put the raw price data and perhaps additionally to some other data or formulas in other columns.

When EleExxel generates the output MSeExcel sheet it will place the raw price from the database into the cells of column E and therefore the formulas, which you have previously placed in the cells of column D, will now have some real values that they can refer to. Hence, they will calculate and hold the result of their calculation based on the content of the output data from column E and, if needed, of additional formulas in other columns outside the print area.

All you have to do is hide the columns with the raw price and some other calculation data from the output by correctly defining the print area.

Calculating prices in and by template files

When generating product lists you can use the calculation functionality of EleExxel to individually calculate prices. You can do so by adding formulas to the non-printable area of the template sheets. These formulas refer to cells, which are filled by EleExxel from Elestore content. The result of these calculations should then be in the printable areas so that the result is calculated by the formulas using the base data from the Elestore database.

Although this form of calculating prices is substantially simpler than the possibilities offered by Elecalc you can still apply simple and very effective calculations to your output MSeExcel sheets generated by EleExxel.

Using cell references

Please remember that MSeExcel offers many possibilities of referencing other cells directly or indirectly. Like this you can fill the cells of the output area depending on the values and content of some other cells located outside the print area.

This is how you can create very flexible and dynamical EleExxel template sheets. How you do this depends primarily on your intentions and on the question, which part of the template content is to be changed under which conditions.

If you are not an expert in MSeExcel yourself and don't have one at hand you can address yourself to our support who will solve your individual questions and problems or even construct individual EleExxel template sheets for you.

Create a non-repeatable template from the output of a repeatable template

Make a repeatable template with the goal to generate article numbers (keys) to be used for creating a non-repeatable template file. The output of the repeatable template then holds the article numbers, which you need in the new non-repeatable template file. All you have to do is rearrange these article numbers as needed and that is certainly quicker and less typo-error prone than copy/pasting or even typing in the article numbers into the new non-repeatable template.

Avoid language dependent text in EleExxel template files

In most cases some text will have to be directly included in template files, which is not imported from data objects. This is often the case for titles or footnotes. Such texts inside a template file make this template file language dependent. Thus, you would need different template files for every language and that is not a good solution as it creates data redundancies and unnecessary work.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

However, you can easily avoid this by creating some fake article objects, which only exist to act as holders for such language dependent texts. You then include references to these special fake objects in order to make your template files fully language independent.

Note: This manual is "work in progress" and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Interaction problems MSOffice with EleFamily

Important "don'ts" when using EleFamily with MSOffice

Needed MSOffice versions

First of all, your MSOffice software must be of version 2002 or newer. Older versions of MSOffice do not support the interface, which is needed and used by the EleFamily software to drive the MSOffice software.

Therefore, older versions of MSOffice cannot be used and this is not our fault!

MSWord and MSeExcel

- Never open MSeExcel while EleExel is running. This will lead to uncontrollable results and most likely EleExel will crash. There is no way how we can prevent this. Just DO NOT WORK with MSeExcel while EleExel is generating.

The same applies for ElePricelist to MSeExcel and also to MSWord! DO NOT USE THEM while ElePricelist is running! And because EleProductSite uses EleExel and/or ElePricelist, this also applies to EleProductSite!

- Never click into an MSeExcel sheet, which is automatically opened and processed by our software. This will most likely lead to a crash, which we cannot prevent. Don't do the same to an MSWord text, which is opened and processed by our software.
- All of these issues also apply to MSWord accordingly.

Interaction between EleFamily and third party software

When one of the EleFamily products uses MSeExcel, MSWord or third-party software the EleFamily software to a certain extent "drives" the other software that means that the EleFamily software operates the third-party software similar to an end user operating on the computer keyboard.

However, it is technically not possible that our software gains full control and or total "authority" over the third party software. In practice that means that quite easily situations can occur where the third-party software like MSOffice does not completely react to the commands received from our software as expected. In most cases these unexpected reactions of the third-party software are due to some settings or conditions of this particular installation, which our software cannot influence or modify.

If such a situation occurs we ask you not to blame the EleFamily software product, simply because our software has in most cases no chance to overwrite the behaviour of the third-party software including MSOffice. Of course, we will try to help you as good as we can to overcome any such problem but we cannot guarantee that this will always succeed.

In this chapter we give you some issues, of which we know that they can cause problems. Please try to use this as a checklist when you are trying to find the reason for some unexpected behaviour.

Macros in MSeExcel or MSWord

We found out that quite frequently macros in the file "normal.dot" or in another format file, which is used and referenced, cause problems. Please remove all macros from the "normal.dot" or other such files when using EleFamily.

Language settings in MSOffice

Crashes occur regularly if the language setting of your MSOffice software is different from the language setting of your Windows operating system. This is obviously a bug in MSOffice software and, of course, we have no possibility of removing this bug.

You must use the same language setting in your MSOffice software, which has also been selected for your Windows operating system.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

ElePricelist

Overview on ElePricelist

ElePricelist brief abstract

ElePricelist allows the export of data from Elestore to well-formatted MSWord documents. ElePricelist can also embed MExcel tables into these MSWord documents. These MExcel tables are generated by EleExcel, which is fully incorporated in ElePricelist and acts under control of ElePricelist.

ElePricelist generates MSWord documents from user-defined templates with or without embedded MExcel files that are generated by EleExcel.

Mixing automatically generated MExcel and MSWord files and merging them into one holding MSWord document enables extremely flexible design and document generation capabilities from simple user-defined templates files.

ElePricelist fully incorporates EleExcel as well as the Elestore database.

The ElePricelist software is available in different forms:

- As stand-alone ElePricelist software product incorporating a basic version of the Elestore content database as well as the complete EleExcel product, which is always part of ElePricelist
- As an optional module extending Elestore
- As a combined product together with Elecalc, Elequote and/or other members of the EleFamily
- As an integral part of EleProductSite, which includes the full ElePricelist and the full EleExcel products.

The Elestore user manual is needed as an extension of this ElePricelist manual just as the Elestore software and database are integral part of ElePricelist.

The differences between ElePricelist and EleExcel

When comparing ElePricelist with EleExcel the primary advantage of ElePricelist is in using MSWord, which offers far more flexibility in designing and formatting output due to the far superior formatting capabilities of MSWord versus MExcel.

Due to MSWord's capability to include MExcel sheets that were generated by EleExcel into MSWord files ElePricelist offers far greater flexibility in creating individually designed and automatically generated output from content in Elestore.

Differences between ElePricelist and EleExcel related to MSOffice

EleExcel is based on MExcel while ElePricelist uses MSWord for generating output. The fact, that ElePricelist includes MExcel tables, which are generated by EleExcel, does not make a substantial change, because these MExcel tables are included as a static block into MSWord.

Due to all of this there are some differences between ElePricelist and EleExcel, which are consequences of the differences between MSWord to MExcel:

- MSWord has no printable area and therefore no area, which is not printed.
- MSWord primarily relies on floating text, which is formatted by MSWord.
- The tables in MSWord have a substantially different behaviour compared to tables in MExcel.

References to the EleExcel user manual

Due to the many similarities between ElePricelist and EleExcel we intentionally do not replicate those parts of the EleExcel user manual, which are related to features that are also supported by ElePricelist. In these cases we simply refer to the respective chapter in the EleExcel manual.

As soon as EleHelp is fully functioning, we will intentionally include the respective chapters redundantly as some good examples how redundant output is created by EleHelp without having to create and store redundant content.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

ElePricelist extends EleStore and EleExxel

ElePricelist is based on the Elestore content database and it substantially extends EleExxel:

- ElePricelist generates MSWord text files with individually formatted complex layout structures takes from user-defined MSWord template files, which are automatically filled with content from the Elestore product database
- ElePricelist generates well-formatted, multi-line MSeExcel sheet from data in Elestore (it includes EleExxel)
- ElePricelist generates individually formatted product pages in HTML or even entire product websites including the navigation content.

The practical usage of ElePricelist manifold:

- Generate printable, well formatted MSeExcel sheets from Elestore content as a simple form of database publishing
- ElePricelist generates nicely formatted and individually calculated price lists in MSeExcel or MSWord
- Generate product lists in HTML with links to detailed product pages in HTML and/or to the EleAIS database.

ElePricelist makes the most of MSeExcel and MSWord and combines both

MSWord has even better formatting and printing as well as very comprehensive exporting capabilities compared with MSeExcel. Almost every computer user knows how to use MSWord and MSeExcel and these software products are on most computers anyway.

This is why ElePricelist provides comfortable and comprehensive mechanisms for using MSWord and MSeExcel and their features as a generator for the data stored inside the Elestore database. ElePricelist fully automatically uses and drives MSWord and MSeExcel as an output and print generator as well as an export manager supporting most physical data formats in use today.

The standard Elestore software already fully embeds MSeExcel both as a QuickEditing tool as well as an export target for data from the Elestore database. This has been available since one of the first Elestore releases way back in early 2004.

ElePricelist extends this by providing even far more comprehensive capabilities in using MSWord and MSeExcel especially for formatting multi-line data output from the Elestore database. ElePricelist provides simple, quick, yet very powerful features for using MSWord and MSeExcel as well as both together as universal output and print generators and as product and price list generators for data in Elestore

How ElePricelist is used

For the basics please refer to the manual on EleExxel, because they are the same for ElePricelist. Here we only explain the subjects in which ElePricelist extends EleExxel or differs from EleExxel.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

ElePricelist templates – structure, format, data references

What are ElePricelist templates?

ElePricelist uses templates in the format of MSWord files (*.doc). These are the blueprints, the construction plans for ElePricelist to generate MSWord output files. These templates tell ElePricelist where to place what kind of data and they also define if and where MSeExcel files that are generated by EleExxel are to be placed into the output file.

This chapter discusses the elements and the structure of ElePricelist template files and how data is referenced in templates.

References to data in ElePricelist templates

Overview on the purpose of referencing content in ElePricelist template files

ElePricelist combines the formatting power of MSWord with the many features of MSeExcel and this amalgam is a very powerful tool to publish data in Elestore formatted by user-defined templates.

In ElePricelist you can either solely rely on MSWord for formatting your data output or you can embed MSeExcel files, which were generated by EleExxel under the control of ElePricelist. For all of the details concerning EleExxel please refer to the EleExxel user manual. This is not repeated here.

Static data in ElePricelist templates mixed with automatically generated data

Beyond the already very powerful page generation capabilities of ElePricelist and the embedded EleExxel there is another great advantage of ElePricelist using MSWord and that is in cases where comprehensive static text is to be enhanced and merged with dynamically generated database content.

This is a task that ElePricelist solves perfectly!

You can extremely well and very easily embed references to Elestore database content in static text in an MSWord template and the references to the Elestore data objects will be filled with content during the generation run. This is how you can very easily individualise documents of different types, offers, price lists, advertising and marketing paper and many other types of documents, which exist of a static and generic part and individual inserts, which are added by the ElePricelist software from data in Elestore.

Referencing data attributes (fields) in ElePricelist template files

Some text chunk in an ElePricelist template file, which is to be filled by ElePricelist with some data from the Elestore database, must contain a reference to an attribute of the article object class in Elestore. Please note that we are using the term “attribute” while the terms “data field” or “variable” are also frequently used. In most cases they can be used synonymously.

The attribute, which is referenced, is identified by the attribute name as used in the Elestore database. Please refer to the Elestore user manual for details how attributes are named in Elestore. We use a very simple but concise and obvious naming convention.

Attributes in ElePricelist templates are always referenced directly. That means that the attribute name is stored directly inside the ElePricelist text or inside a MSWord table cell, which is supposed to be filled with the value of this attribute. See below for the syntax.

Indirect references are only available in EleExxel, because of the nature of MSWord, which does not support this and which does not support non-printable areas.

Referencing data objects (records) in ElePricelist template files

Data objects are referenced only inside non-repeatable ElePricelist templates.

In non-repeatable templates the identification of the data objects, whose data is to be output, must be stored directly inside the respective text.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Defining the exported attributes (data fields) in ElePricelist

The attributes of the article objects, whose values are to be exported by ElePricelist, are defined in the template files by VariableTemplates. In this respect the way that ElePricelist exports data from Elestore is very much different from the normal export manager of Elestore.

The way ElePricelist references attributes is very much similar to EleExcel with the exception that indirect references to attributes are not available, of course, because MSWord tables do not support references sufficiently like MSExcel does. Of course, attribute references can also be placed inside normal floating text in an ElePricelist template document.

Direct attribute references only

An ElePricelist template file can only contain direct references attributes. The name of the referenced attribute of the data object is contained directly in the text or in the MSWord table cell, which is supposed to be filled with the value of this attribute during the generation process.

This is a sample referring directly to an attribute:

#art00020PrimExtNr

This is explained in more detail further below.

Syntax of direct attribute references in ElePricelist templates

Attributes are referenced by the name of the attribute preceded by the hash sign “#” (without quotes). The syntax of direct references must be used as shown in this example: #art00020PrimNrExt

In this case the actual name of the attribute is "art00020PrimNrExt" (note that the quotes are not part of the name of the attribute). The hash sign (“#”) must always be added as the first character of the reference to an attribute.

This means that the whole attribute reference is always composed of the hash sign as the first character plus the name of the attribute, which must be correctly spelled and without any other character between the hash and the attribute name. Here is an example with three references, which we have embedded into a simple table to simulate a small MSWord table but the same can also be placed inside normal floating text:

#art00020PrimExtNr	#art05011TextShort1	#art07511Length
--------------------	---------------------	-----------------

- The first cell contains a reference to the primary external article number, which can be used as the key to identify an article object.
- The second cell contains a reference to the field article text 1 of the article object. This is a string field.
- The third cell of this example contains a reference to a numeric field, which contains the value for the length.

In all of these cases the cell holding the attribute reference is filled by ElePricelist during the generation process with the value of this attribute from the currently processed article data object. The cell indicates which attribute value to use. The same applies, of course, if the attribute reference is not contained in a table cell but just somewhere in some floating text.

Defining the exported data objects (instances, records) in ElePricelist

Overview on defining the exported data objects

It depends on the type of template how and where the data objects are defined, which are to be exported:

- In the case of a repeatable template the data objects are selected by the user in one of different ways in the project form by performing some search, by selecting specific objects or one of the other ways. Please refer to the chapter about the ElePricelist project settings for a full description.

In some versions of Elestore the exported data objects can also be defined by so-called StoredSelections. They are permanently stored, user-defined selections (data sequences) of any number of data objects in any specific order.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

In any of these cases a collection of article objects as defined by the user is used to provide the ElePricelist generator with the data objects for the output.

- In the case of non-repeatable templates the data objects to be exported are referenced directly in the non-repeatable template document. These data objects are defined by one of their article numbers (primary keys).
- In the case of ElePricelist using EleExxel projects both variants can be freely mixed with one another.

Referencing data objects (records) in repeatable and non-repeatable ElePricelist template files

The use of references to data objects depends on the type of the ElePricelist template file. In this respect we have to clearly distinguish between repeatable and non-repeatable template files.

[References to data objects in repeatable ElePricelist templates](#)

First of all, we have to recognise that data objects are never referenced in a repeatable ElePricelist template file. This is the case, because it is a fundamental feature of repeatable ElePricelist template files that the data objects, whose data is to be filled into these templates, are determined by the user prior to starting the ElePricelist generation process. The data objects for repeatable template files are searched, selected or defined by the stored selection. In other words, there are never any references to data objects in repeatable templates, because they are added automatically by ElePricelist.

[References to data objects in non-repeatable ElePricelist templates](#)

Non-repeatable ElePricelist template files must contain references to data objects to work properly. Please read the following chapter about the syntax of references to data objects where the syntax of these references is explained.

StoredSelections for exporting data objects

Please refer to the EleExxel manual.

Syntax of data object references in ElePricelist templates

Non-repeatable templates must also contain links to the data objects, whose values are supposed to be filled into the template document. Therefore, non-repeatable templates must contain references to attributes as well as references to data objects. Both references are included in one link.

[Syntax of the combined reference to data object and attribute](#)

Here is an example:

```
<!EME X15-320;#art00020PrimExtNr>
```

The reference consists of several components, which are explained from left to right:

- The reference must start with the characters “<!” (as always without the quotes).
- The next part consists of the primary key of the data object, in this case “EME X15-320”
- Next part is the delimiter character “;” (semicolon), which separates the reference to the object in the left part from the reference to the attribute.
- Next part is a normal reference to an attribute with the hash sign (“#”) as shown above in the first row or it’s an indirect reference, i.e. a link to another column as shown above in the second row.
- To close the reference there must be the “>” at the end.

For further examples please refer to the sample template files, which are shipped with ElePricelist.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Special handling of dynamic attributes in ElePricelist

For the general information about dynamic attributes please refer to the EleExxel manual.

Example: Special handling of dynamic attributes in ElePricelist

Here is a symbolic sample excerpt from an ElePricelist template document file using an MSWord table:

A	B	C	D	E
Data04	#DynName01	#DynValue01	#DynName02	#DynValue02
Data05	#DynName03	#DynValue03	#DynName04	#DynValue04

Column A indicates the purpose of these template rows; in this case these rows are to contain data fields. Columns B and D hold templates for the names (labels) of dynamic attributes. Columns C and E hold templates for the values of dynamic attributes.

After being filled with data the output rows created from these template rows might look like this (column A has been made invisible, because it's not wanted in the output):

B	C	D	E
Height	100 mm	Width	150 mm
Colour	Blue	Weight	1200 g

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

ElePricelist repeatable templates

Overview

For an explanation of repeatable and nonrepeatable templates please refer to the user manual on EleExcel. These subjects are essentially identical in ElePricelist.

This chapter covers only the MSWord templates used by ElePricelist. It does not cover the features of EleExcel, which are embedded in ElePricelist. For the MSExcel related parts and templates please refer to the user manual on EleExcel.

What means “repeatable template”?

When using "repeatable templates" ElePricelist takes one MSWord template document and duplicates it as many times as there are article objects to export. Each of these repeated copies of the template document is filled with the data of one article object. As a consequence, each of these copies of the template file looks fairly identical in their layout although each of them contains different data.

This one repeatable template document therefore defines the layout and the formatting of all exported data objects.

Repeatable templates contain only links to data fields but no references to data objects whose data is inserted, because the data objects (instance or records) are defined by searches, selections or externally by the operator.

This form of using ElePricelist is the preferred one when a relatively large amount of data objects is to be generated fully automatically in a fairly uniform layout. Creating this one repeatable template is much simpler and quicker compared to the non-repeatable template type.

Any MSWord user can do. It's easy, simple, quick, yet extremely powerful.

Structure of an ElePricelist repeatable template

Sample ElePricelist template MSWord file

This is the sample template file, which is discussed throughout this chapter. Your templates can look differently, of course.

<#art00020PrimNrExt> <#art00030Name> <#art07101VK01>

<#art06060Comm01>
<#art06063Comm04>
<#art06061Comm02>

Technical table:

Weight net.	Length in mm	Units of quantity
<#art03010WeightNet>	<#art03051Lenth>	<#art75178UserAttribute>

[Explanation of a repeatable ElePricelist template file \(sample\)](#)

This sample template document starts with a title paragraph that consists of the three attributes, which are to be printed at the top, distributed over two paragraphs. The first attribute (art00020PrimNrExt) contains the article number and the second one (art00030Name) the product name.

The third attribute reference is contained in the second paragraph, which is right justified, and it refers to one of the several price attributes of an article object.

Of course, choosing these attributes has to correlate with the previous decision to store the respective values in just these attributes. For the product name and the price this could also have been one of the many other article attributes (string fields or prices).

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Therefore, such attributes in the template must correspond to those where you have stored the values in Elestore.

Below the title paragraphs there are three rows in an MSWord table, which contain general technical descriptions of this article.

The bottom part of this template is consumed by an MSWord table which contains references to different attributes. The second row contains references to different attributes of the article object, whose data is exported into this sequence.

When the project, which refers to this template, is executed, the entire template is duplicated as many times as there are article objects to process. The duplicated copies of the template are filled with data from the respective article objects and the result is added to an output MSWord document or each one is stored as a separate file depending on the settings and the version of ElePricelist.

This is why that is called a repeatable template.

No row identifiers in ElePricelist

In EleExcel the structure of MSExcel spreadsheet templates relies to a large degree on row identifiers (for details see the EleExcel user manual). Such row identifiers do not exist in MSWord and therefore not in ElePricelist, simply because they would not make any sense, not even inside MSWord tables, which behave very much differently from an MSExcel spreadsheet.

Another reason is that MSWord does not offer any features similar to the printable area in MSExcel, which also implies that there is a non-printable area in MSWord, except if a style with a hidden font is used but that is something different. All areas in MSWord are printed and that is another reason why row identifiers are not possible in ElePricelist.

No comment or nop rows in ElePricelist

As a consequence, you cannot include comment or no operation ("nop") rows into an MSWord template file. But in practice this is a completely unimportant little disadvantage. The many advantages of the much greater formatting capabilities outweigh this by far.

Explanation of the ElePricelist template MSWord file (sample)

This is a discussion of the structure of this sample ElePricelist template file. You find a more detailed discussion on the purpose of the various elements further below in this manual.

Static attributes in the sample template file

The first rows refer to static attributes. The values in these template cells are VariableTemplates, which are explained in the chapter about EleExcel.

Dynamic attributes in an ElePricelist template file

Attributes are referenced in cells via this syntax: <#DynValueXX> where “XX” refers to a number that relates to the sequence of an attribute inside the article object. Attribute names are referenced through this syntax: <#DynNameXX>

eCl@ss variable in the sample template file

The cell with the VariableTemplate “BAA27101”, which is taken from the eCl@ss attributes. This is explained in detail in the Elestore user manual in the chapter about the eCl@ss product classification system and its defined attributes. The syntax to reference this variable is <#BAA27101>.

Dynamic attributes in the sample template file

The bottom rows of the second table refer to dynamic attributes, which are in this case grouped in two logical columns.

The values in these template cells are dynamic VariableTemplates, which are explained in the chapter about EleExcel.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Sample output of a repeatable template for ElePricelist

[Result in MSWord](#)

This is a part of the MSWord file, which was created from the above template sample with real data:

100- 4x5 Double Open-Ended Spanner

3.70

Chrome-Vanadium 31 CrV 3, Code No. 1.2208		
DIN 3110/ISO 3318/ISO 1085		
jaws set at 15°, chrome plated		
Technical table:		
Weight net.	Length in mm	Units of quantity
16	110	12

100- 4,5x5,5 Double Open-Ended Spanner

3.40

Chrome-Vanadium 31 CrV 3, Code No. 1.2208		
DIN 3110/ISO 3318/ISO 1085		
jaws set at 15°, chrome plated		
Technical table:		
Weight net.	Length in mm	Units of quantity
16	110	12

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

[Result in HTML](#)



100- 4x5 Double Open-Ended Spanner

3.70

Chrome-Vanadium 31 CrV 3, Code No. 1.2208

DIN 3110/ISO 3318/ISO 1085

jaws set at 15°, chrome plated

Technical table:

Weight net.

16

Length in mm

110

Units of quantity

12

100- 4,5x5,5 Double Open-Ended Spanner

3.40

Chrome-Vanadium 31 CrV 3, Code No. 1.2208

DIN 3110/ISO 3318/ISO 1085

jaws set at 15°, chrome plated

Technical table:

Weight net.

16

Length in mm

110

Units of quantity

12

The look of the HTML file is very similar to the original in MSWord (The image size was intentionally shrunk).

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

ElePricelist non-repeatable templates

What means “non-repeatable template”?

ElePricelist non-repeatable template documents are identical to EleExxel non-repeatable templates with the difference that ElePricelist uses MSWord, of course, while EleExxel uses MSeExcel. Therefore, please refer to the EleExxel manual for a general description of non-repeatable templates, because most of that is also valid for ElePricelist.

Discussion of a non-repeatable template for ElePricelist

Here is a view on a sample ElePricelist template document (further below you see a final document generated from this template):

<100;#art00020PrimNrExt>

<100;#art00030Name>

<100;#art05311Image01>

Material	<100;#art06060Comm01>
Type	<100;#art06061Comm02>
Standard	<100;#art06063Comm04>

Article number	Size in mm	Length	Units of quantity	Weight of each piece	Price
<100-4x5;#art00020PrimNrExt>	<100-4x5;#art75212UserAttribute>	<100-4x5;#art03051Lenth>	<100-4x5;#art75178UserAttribute>	<100-4x5;#art03010WeightNet>	<100-4x5;#art07101VK01>
<100-4,5x5,5;#art00020PrimNrExt>	<100-4,5x5,5;#art75212UserAttribute>	<100-4,5x5,5;#art03051Lenth>	<100-4,5x5,5;#art75178UserAttribute>	<100-4,5x5,5;#art03010WeightNet>	<100-4,5x5,5;#art07101VK01>
<100-5,5x7;#art00020PrimNrExt>	<100-5,5x7;#art75212UserAttribute>	<100-5,5x7;#art03051Lenth>	<100-5,5x7;#art75178UserAttribute>	<100-5,5x7;#art03010WeightNet>	<100-5,5x7;#art07101VK01>

Further below you find some final results, which were generated from this sample file.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

[Explanation of a non-repeatable ElePricelist template file \(sample\)](#)

This sample template document starts with a title paragraph that consists of the two attributes, which are to be printed at the top. The left attribute (art00020PrimNrExt) contains the article number and the right one (art00030Name) the product name. In both references the article number (ident number) “100” is used in this case. Alternatively, there the external article number could also be used instead.

Of course, choosing these attributes has to correlate with the previous decision to store the respective values in just a these attributes. At least for the product name this could also have been one of the many other article attributes (string fields). Therefore, such attributes must correspond to those where you have stored the values in Elestore.

Below this there is a large frame, which takes up an image of this article.

Below the image there are three rows in an MSWord table, which contain general technical descriptions of this article.

And the major part of the template (only the first three rows are shown in this case) is consumed by an MSWord table which contains references to different attributes and different articles. As you can see the first column contains different references in this case to child articles of the article object, which is referenced in the top paragraph (Article 100).

[Discussion of the references of a cell](#)

As you can see most of the cells contain a formula like one of the following:

<100;#art00020PrimNrExt >

<100-4x5;# art00020PrimNrExt >

The left part of this formula, which represents the identification of an article object, refers to the article attribute, which identifies one particular article object and that is one of the two primary article numbers. Remember, that we have a non-repeatable template in this case and that therefore the data object identifications must be contained in the template.

The left part of the formula with the object identification is separated by the delimiter character semicolon (“;”) from the right part of the formula, which contains the attribute identification.

The second part of the formula represents the attribute (data field) and that is in this case the external article number in the field art00020PrimNrExt.

[Font definition inside the template](#)

As you can see very well especially from the top paragraph and also from the first table column with the article children of this example the formatting can be freely applied to the template and this formatting is used by ElePricelist accordingly for the generated output pages, provided that this is activated by the corresponding setting in the project.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

View on a the final page created from a non-repeatable ElePricelist template file

This is a page, which was generated automatically by ElePricelist from the above shown non-repeatable template:

[View of a sample page in MSWord](#)

This is a part of the MSWord file, which was created from the above template sample with real data:

100 Double Open-Ended Spanner



Material	Chrome-Vanadium 31 CrV 3, Code No. 1.2208
Type	jaws set at 15°, chrome plated
Standard	DIN 3110/ISO 3318/ISO 1085

Article number	Size in mm	Length	Units of quantity	Weight of each piece	Price
100- 4x5	4x5	110	12	16	3.70
100- 4,5x5,5	4,5x5,5	110	12	16	3.40
100- 5,5x7	5,5x7	130	12	24	3.85
100- 6x7	6x7	130	12	24	3.39
100- 6x8	6x8	130	12	24	3.88

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

[View of a sample page in HTML](#)



100 Double Open-Ended Spanner



Material	Chrome-Vanadium 31 CrV 3, Code No. 1.2208
Type	jaws set at 15°, chrome plated
Standard	DIN 3110/ISO 3318/ISO 1085

Article number	Size in mm	Length	Units of quantity	Weight of each piece	Price
100- 4x5	4x5	110	12	16	3.70
100- 4,5x5,5	4,5x5,5	110	12	16	3.40
100- 5,5x7	5,5x7	130	12	24	3.85
100- 6x7	6x7	130	12	24	3.39
100- 6x8	6x8	130	12	24	3.88

The look of the HTML file is very similar to the original in MSWord (The image size was intentionally shrunken).

More templates and sample pages on our websites

Please look on our product websites for more template samples and for pages generated by these samples. Please refer to www.elestore.com or www.elestore.de in Deutsch and in early 2007 also to www.elepricelist.com (still in preparation).

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

ElePricelist embeds EleExxel and ElePricelist projects

Overview

Probably the greatest advantage and power of ElePricelist is in its capability to include several EleExxel and ElePricelist projects. In this respect one ElePricelist project acts as the master of several or many other EleExxel or ElePricelist projects whose output and results are included in the document created by the master ElePricelist project.

One master ElePricelist project can embed several other EleExxel and/or ElePricelist projects. This master project also automatically executes all embedded dependent projects and takes care that their results are added to the master's output document.

This feature of ElePricelist allows you to create an output MSWord document from several or many embedded MSeExcel and/or MSWord documents, which were created by separate and independent EleExxel and/or ElePricelist projects.

This feature of ElePricelist also enables you to re-use other projects, which can also be used autonomously when needed.

This chapter explains ElePricelist templates, which embed other EleExxel and/or ElePricelist projects.

Syntax for embedding other projects in an ElePricelist master project

There is a very simple syntax, which embeds other EleExxel and/or ElePricelist projects. You simply add a command with the following syntax to the master ElePricelist template file in MSWord:

<!project;5>

Explanation of the syntax of the ElePricelist project command

The term “project” (no quotes) tells ElePricelist that it is supposed to execute another EleExxel or ElePricelist project and to include its output result into a copy of the ElePricelist template at this location.

The command "project" is followed by a colon, which acts as a separator.

On the righthand side of the colon there is the ident number of the project, which is to be executed and whose result is to be included at this position. The ident number is the unique identifier of a project.

The ident numbers are the primary and unique identifications of projects similar to all other data objects in EleFamily. There is no distinction between the ident numbers of EleExxel and ElePricelist, because the projects for both software-products are stored in the same database table.

Therefore, ElePricelist can easily detect from the data of the referenced project whose identity number is included in this command if this is an EleExxel or ElePricelist project. This is also very easy to determine for you, because the ident numbers of the projects are shown in the selection forms both of EleExxel as well as of ElePricelist.

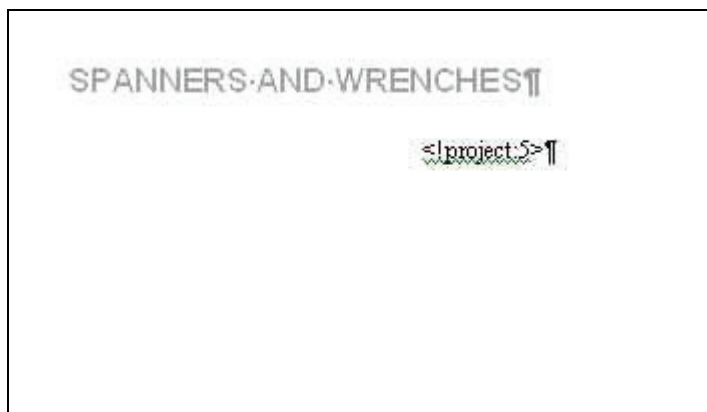
Of course, one would typically include several or many other projects into one such ElePricelist master template.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Structure of an ElePricelist embedded EleExcel template

Sample ElePricelist template MSWord file


This is the sample template file:



The border around this screenshot has been set only to better visualize the area covered by the screenshot.

[Sample result created by an embedded ElePricelist project](#)

SPANNERS AND WRENCHES

894 Single-Open-Ended-Spanner				
				
Type	high-grade tool steel			
Material	according to DIN 894/ISO 3318			
Standard	jaw set at 15°, black finish			
Stock-No.	T mm	w	W g	€
894-5,5	77	15	12	7,20
894-6	77	15	12	3,00
894-7	82	15	15	2,98
894-8	97	15	24	2,94
894-9	102	15	30	3,40
894-10	107	15	31,5	3,19
894-11	117	15	40,5	3,19
894-12	127	15	45	3,30
894-13	132	15	54	3,42
894-14	142	15	63	4,26

You do not see any relation between this output and the template, which was in charge of creating the above document in MSWord. The reason for this is in the fact that the ElePricelist template called another project and all of the information and formatting, from which this output was created, is contained in this referenced project.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

ElePricelist projects and their settings

Overview and purpose of projects in ElePricelist

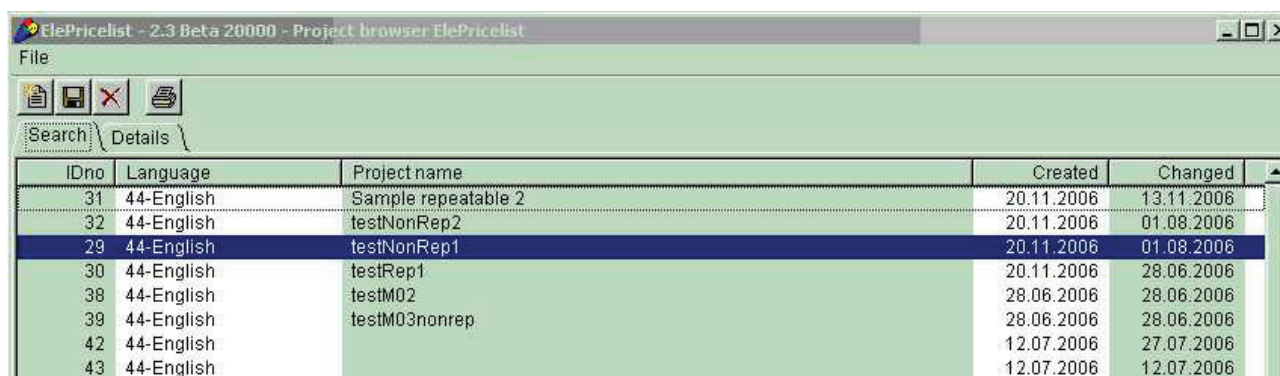
There are various settings, which decide about how the ElePricelist generator behaves when creating output and when adding values to the cells of your ElePricelist template. Some of these settings substantially influence the content and/or the look and feel of the generated output document.

All of the parameters and settings, which are to be used by ElePricelist for running a job, are entered and stored in an ElePricelist project for quick re-use.

We have implemented all settings and rules, which we thought to be of practical advantage for you as an ElePricelist user. However, ElePricelist is a relatively young product (first published in I. Q. 2007) and has therefore not yet been influenced too much from end-user's experiences. We invite you to please let us know your requirements for future releases of ElePricelist and we will be happy to implement such requirements by adding additional settings and rules in order to improve the usability of future ElePricelist releases. Please let us know your demands and requirements!

ElePricelist project table

This is a partial view of the project selection table of ElePricelist:



IDno	Language	Project name	Created	Changed
31	44-English	Sample repeatable 2	20.11.2006	13.11.2006
32	44-English	testNonRep2	20.11.2006	01.08.2006
29	44-English	testNonRep1	20.11.2006	01.08.2006
30	44-English	testRep1	20.11.2006	28.06.2006
38	44-English	testM02	28.06.2006	28.06.2006
39	44-English	testM03nonrep	28.06.2006	28.06.2006
42	44-English		12.07.2006	27.07.2006
43	44-English		12.07.2006	12.07.2006

You can select any ElePricelist project from this table for simple re-use. Click on one of the existing projects and then switch to the details form to re-use the stored project with or without modification of the settings.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

The ElePricelist project details form

View of the ElePricelist project details form

The screenshot shows the 'ElePricelist - 2.3 Beta - Project browser ElePricelist' window. The 'Details' tab is active, displaying various settings for a project. The settings are organized into sections: 'Settings', 'Handling of page breaks', 'Handling of formatting', and 'Handling of auto-fit'. The 'Settings' section includes fields for Project name, Name of template file, Type of template, Export type, Office product group, Name of output file, Primary key to use, and Language. The 'Handling of page breaks' section includes fields for Page break after... and Page break after number of... The 'Handling of formatting' section includes a checkbox for Use formatting from MSWord. The 'Handling of auto-fit' section includes checkboxes for Auto fit text objects, Auto fit pictures, Lock aspect ratio for texts, and Lock aspect ratio for images. At the bottom of the window are buttons for Cancel, Select articles, and Export.

Section	Setting	Value	Options
Settings	Project name	100(rep)	
	Name of template file	..Templates\ElePricelistRepeatable\Template-reals100(rep).doc	...
	Type of template	10-Repeatable template (duplicate one template many times)	...
	Export type	200-Export all of a grouping level	...
	Office product group	4-100	...
	Name of output file	..ElePricelist100(rep).doc	...
	Primary key to use	10-Ident no. (art00010IdentNr)	...
	Language	44-English	...
Handling of page breaks	Page break after...	20-Insert page break after number of articles	...
	Page break after number of...	5	
Handling of formatting	Use formatting from MSWord	<input type="checkbox"/>	
	Handling of auto-fit	<input checked="" type="checkbox"/> Auto fit text objects <input checked="" type="checkbox"/> Lock aspect ratio for texts <input checked="" type="checkbox"/> Auto fit pictures <input checked="" type="checkbox"/> Lock aspect ratio for images	

This is the ElePricelist project details form with the settings.

Menu and buttons in the ElePricelist project settings form

[File menu in the ElePricelist project settings form](#)

The file menu contains the usual menu items for adding, deleting, saving, or printing a project.

[Button Select articles in the ElePricelist project settings form](#)

Pressing this button opens the form for selecting article objects. Please refer to the EleExel manual for a detailed discussion of the article selection form.

[Button Export in the ElePricelist project settings form](#)

Pressing this button starts the processing of this project. If you have not yet selected the article objects the form for selecting article objects will be opened first.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

The settings of an ElePricelist project

Project name

This is the same as in EleExxel. Please refer to the EleExxel manual.

Name of the template file

This is the same as in EleExxel. Please refer to the EleExxel manual.

Type of template

Selecting the type of template is the most important of all settings. The type of template selected here must correspond to the content and format of the template file, which you select in the field name of template file.

Please note that there is no possibility for ElePricelist to check whether the type and the structure of a template file corresponds to the selected type of template.

Please read carefully the description of the different types of templates before selecting a type of template.

Trial version:

Users of the trial version are recommended to use one of the predefined projects where the correct type of template should be defined according to the sample template files delivered with the trial version. If you change the type of template or the template file in a project you may get unexpected results if the template file does not correspond to the selected template type.

Concerning results from EleExxel, which are included in an ElePricelist output, this is the same as in EleExxel. Please refer to the EleExxel manual.

[Repeatable template \(duplicate one template many times\)](#)

This is the same as in EleExxel and applies also to the MSWord files of ElePricelist. Please refer to the EleExxel manual.

[Non-repeatable template \(fill all in one template\)](#)

This is the same as in EleExxel and applies also to the MSWord files of ElePricelist. Please refer to the EleExxel manual.

[Autonomous pages \(one template per page\)](#)

This is the same as in EleExxel and applies also to the MSWord files of ElePricelist. Please refer to the EleExxel manual.

Output of single pages to HTML

the output generated from single page templates to HTML is not available in all versions of ElePrice-list. This feature is one of the primary features of EleProductSite.

This templates type is not available in all versions of ElePricelist.

With inheritance

This is the same as in EleExxel. Please refer to the EleExxel manual.

Export type

This is the same as in EleExxel. Please refer to the EleExxel manual.

Office product group

This field is only available if you have selected in the field type of template the type 200, that is export all of a grouping level. In all other cases this field is disabled.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

This field allows you to select a grouping level from the office grouping system (please note that there are typically several different grouping systems available in ElePricelist). By selecting a specific grouping level you tell ElePricelist, which article objects to process in which sequence, which is both defined by the grouping level and its article objects.

For further details please refer to the chapter about the office grouping system.

Name of output file

Here you can add a file name for the MSWord output file, which is created by the ElePricelist output procedure. By default, Elestore will name the newly created and stored MSWord file using the current date and time in the sortable format ElePricelist_YY.MM.DD_HH.MM.SS.xls.

If you have selected an export type that produces single pages then ElePricelist will use the values of the primary key of the processed data objects to name the created files.

If an ElePricelist project is run under control of EleProductsite then the file name entered here is used and enhanced by sequential page numbers, which are added to this default page name.

Overwrite existing output file

If you activate this checkbox existing files with the same name as the created output file(s) will be overwritten without asking.

Primary key to use

This is the same as in EleExxel. Please refer to the EleExxel manual.

Save also as HTML

If this checkbox is activated ElePricelist will automatically save all generated output files also in HTML format additionally to the MSExcel format (*.xls).

Split HTML pages

If this checkbox is activated ElePricelist will automatically split the exported HTML pages of a repeatable project into single pages. This allows creating single HTML page files from the content of a repeatable MSExcel output sheet.

Language

This is the same as in EleExxel. Please refer to the EleExxel manual.

Apply language on embedded projects

Check this box if you want the language setting also to be applied to embedded projects so that the language setting in these embedded projects is overwritten with this one.

The settings of an ElePricelist project – page breaks

Page breaks after articles

This field is only available if you have selected to use a repeatable template. It would make no sense for non-repeatable templates, because page brakes should be defined within non-repeatable templates themselves.

This setting allows you to tell ElePricelist to automatically add page breaks after a given number of data objects exported to MSWord.

[No page breaks to be inserted](#)

If you select this setting ElePricelist will not include any page breaks into the output file. You can still let MSWord calculate the page breaks itself.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

[Insert page break after number of articles](#)

If you select this setting ElePricelist will include a page break into the output file after every number of articles as specified in the field “Add page breaks after X articles”.

Add page breaks after X articles

This setting allows you to tell ElePricelist to add a page break after any number of exported article objects.

The settings of an ElePricelist project – handling of formatting

Use formatting from MSWord

Is this checkbox is activated the formatting applied to the MSWord template file will be used and copied into the output document. Is this checkbox is deactivated the formatting of the variable reference from the template file will be used. To be precise: The formatting of the first character of the attribute command will be taken and applied to the entire content.

The settings of an ElePricelist project – handling of auto-fit

Auto-fit text objects

If you activate this checkbox ElePricelist will tell MSWord to apply the auto-fit function on all cells holding texts or embedded text objects. This applies only to text boxes.

Lock object aspect ratio for

Text cells

This is the same as in EleExxel. Please refer to the EleExxel manual.

Auto-fit pictures

If you activate this checkbox ElePricelist will tell MSWord to apply the auto-fit function on to all cells holding pictures. This applies only to text boxes.

Lock aspect ratio for pictures

This is the same as in EleExxel. Please refer to the EleExxel manual.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleProductSite

Overview

The ideas behind EleProductSite

EleProductSite is one of several ways to publish the content of an Elestore database on the Internet.

These are these general ideas behind EleProductSite:

- Automate the generation of HTML pages from content in Elestore so that a whole product website can be generated fully automatically.
- Create and organise a publication tree that constitutes the structure of the product website.
- Organise all of the HTML pages that are generated by several or many EleExel and ElePricelist projects so that parts of or an entire publication tree in HTML is created automatically including the needed database entries for EleAIS and/or the data structures to create conventional menus.
- Additionally, HTML pages from other sources like Elepub, Elecat-P2H or manually created HTML pages are included into the publication tree.
- Link HTML pages with database content for use by EleAIS or any other similar webserver software.

EleProductSite re-uses Elestore content on the Internet

The core idea behind EleProductSite is to re-use many of the features of the other EleFamily software products as well as the content of the Elestore database to generate entire product websites in HTML and also the content of a webserver database and the links to and from the HTML pages.

What EleProductSite produces

EleProductSite generates an entire product website from Elestore content with these main components:

- Detailed HTML pages with product data in single pages and/or as product lists
- Overview pages that allow a quick selection of products and that contain links to the details pages
- A webserver database for use by EleAIS or by other webserver applications
- Menu structures providing the navigation in a conventional website running without EleAIS or some comparable software
- All data that is needed to connect the webserver database with the HTML pages and vice versa.

This can be done preferably but not only by using templates in MSeExcel and MSWord format, which can be defined very easily by every user of the MSOffice software and therefore without any special HTML knowledge.

Components of EleProductSite

Essentially, EleProductSite consists of these three components:

- The EleExel product
- The ElePricelist product
- An enhanced grouping system for organising the structure and sequence of the data output of EleProductSite

Additionally, there can be a couple of other modules and features such as the inclusion of the Elepub database publishing software.

Features of EleProductSite

EleProductSite comes with integrated EleExel and ElePricelist

EleExel and ElePricelist are always integrated in EleProductSite. As a technical and logical requirement this is also true for the license and therefore an EleProductSite license always also comprises of one for EleExel and ElePricelist.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleProductSite integrates several other EleFamily products

EleProductSite technically and functionally integrates EleExxel, ElePricelist, Elestore and optionally also Elecat-P2H and Elepub. EleProductSite interacts with these software products and partially executes their projects.

Webserver database and HTML for EleAIS

The data and HTML generated by EleProductSite is either used by the EleAIS webserver to run a product website or it becomes part of your website, which is operated without EleAIS by some other software. In so far, EleProductSite generates all of the data needed (not only) by EleAIS and presents your detailed product spectrum on your website.

EleAIS provides HTML pages for conventional web sites

EleAIS also provides the HTML pages plus the data structures necessary for a navigation menu if your website is run without any database and special application software (“conventional”) like Web shop etc. This enables you to present all of your product data as part of your normal company website but with all details contained in your Elestore database.

For normal MSOffice users

Using EleProductSite to create HTML pages and updates can be done easily by normal EleFamily users without any special knowledge of HTML.

Versions of EleProductSite

EleProductSite is available as an option for Elestore

EleProductSite is either an optional module for an Elestore installation or it comes as a standalone product and then comprises of an Elestore database.

EleProductSite always comes with integrated EleExxel and ElePricelist

If delivered as a standalone product EleProductSite always comprises not only of the basic Elestore software and Elestore database but especially of the entire EleExxel and ElePricelist products. In fact, some HTML related features of EleExxel and ElePricelist are only available if they are used as parts of EleProductSite.

EleProductSite is closely integrated with EleAIS

EleProductSite prepares much of the data, which is needed and used by EleAIS to act as a web server. In so far, EleProductSite is some kind of a back office module whose output and results are needed and used by EleAIS to fulfil its functions.

EleProductSite — EleAIS — CATpool — YellowParts

EleProductSite and EleAIS

In many cases, EleProductSite will closely correspond with EleAIS, the webserver software, which is part of EleFamily. This is why EleAIS is frequently mentioned in this manual on EleProductSite so that you understand the effects that some features of EleProductSite have on the data and the functionality of EleAIS.

This is especially attractive for manufacturers who sell through distributors and wholesalers. Such manufacturers don't want to operate a webshop in order to avoid competition with their resellers. Nevertheless, these manufacturers want to present their entire and detailed product spectrum on their own company website, which is often not yet the case.

This is the type of user that EleProductSite was primarily made for. And this is especially true as EleProductSite is only one of several music instruments for such users in an orchestra of programs that make multiple use of the content in your Elestore database.

EleProductSite with your own website software or webshop

Of course, you can also use EleProductSite to provide the HTML pages and the data for your own individual webserver or webshop and there is no need to operate EleAIS in conjunction with EleProductSite although both products harmonize perfectly with one another.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

[EleProductSite and CATpool](#)

EleProductSite can also be used to provide data and pages for CATpool, the product catalogue and data exchange system. CATpool is member based and the CATpool portal is accessible only by CATpool members.

The parts of CATpool presenting product data and catalogue pages are based on EleAIS and therefore all the statements made with respect to EleAIS also refer to CATpool unless otherwise mentioned.

[EleProductSite and YellowParts](#)

EleProductSite can also be used to provide data and pages for YellowParts, the B2B product and product catalogue portal. YellowParts is a public portal and it is accessible by any anonymous user.

Those parts of YellowParts presenting product data and catalogue pages are based on EleAIS and therefore all the statements made with respect to EleAIS also refer to YellowParts unless otherwise mentioned.

The most important components of EleProductSite

The following list was primarily compiled when looking at EleProductSite as a standalone software product but this is equally true if EleProductSite is installed as an optional module of an Elestore installation:

- As always, an Elestore database and some core Elestore content management software are the most essential parts. Please note that the comprehensiveness of this Elestore content management software will substantially vary between an Elestore installation and a standalone EleProductSite installation.
- EleExcel is always part of EleProductSite so that MSExcel can be used to generate HTML pages.
- ElePricelist is always part of EleProductSite so that MSWord can also be used to generate HTML pages with or without embedding EleExcel projects and output.
- The EleProductSite software itself whose components essentially consist of the following subjects:
 - A special grouping system representing the tree structure of a product website
 - Generator software modules to export this grouping system to the EleAIS database
 - Software to upload content from the Elestore database to the EleAIS webserver database
 - Converter software to import HTML pages generated by Elecat-P2H
 - Software to import the publication structures and pages generated and administrated by Elepub

Static HTML pages versus dynamically generated HTML

EleProductSite intentionally supports static HTML pages, which can easily be made dynamic where really needed. Due to the relatively static nature of product data it is not only simpler and faster but also in many aspects generally better to use static HTML pages.

Static HTML pages have many advantages over dynamically generated HTML pages while dynamically generated HTML pages have only a few and rather rarely applicable advantages over static HTML pages.

Some advantages of static HTML pages

Here is a short and incomplete list of some of the advantages of static HTML pages:

- Static HTML pages are multiple times faster when called from a browser. They are available instantaneously and require much less performance from the webserver.
- Static HTML pages need only a small fraction of the computational power needed by an HTML generator, which generates HTML pages dynamically upon request.
- Static HTML pages are in general much better and simpler detectable by search engine spiders and therefore static HTML pages will be represented much better in Google, Yahoo and the few other important search engines.

There are many more aspects and advantages but this chapter is not an HTML philosophy paper and we therefore keep it at that.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

The website grouping system

Overview

The website grouping system is one of the most important components of EleProductSite and it is the precondition for administrating the structure and the contents of a product website in EleProductSite. Essentially, EleProductSite provides ways of organising the output that is generated by EleExxel, ElePricelist and/or some other programs in HTML.

Organising this output is performed by this website grouping system.

Websites are trees

Every website is constructed as a tree (except those few that consist of one page only).

A tree consists of branches and leafs. The pages of a website constitute its leafs. The branches of the website allow the users to navigate through the tree and to finally select one or several pages.

The website grouping system is represented by the tree of your website.

The structure of this grouping system directly corresponds to the structure of your product website. In fact, by creating the website grouping system and the HTML pages inside it you have already created two of the three essential parts of your product website. The third one is the content of your Elestore database, which is uploaded to the EleAIS webserver database (or some other database). Note that this database is not needed if EleProductSite is supposed to produce HTML pages only without links to some webserver database.

The grouping system also becomes the menu structure

The tree structure of the website grouping system is also exported into JavaScript code, which provides the menu and navigation system for your website if your website is run without EleAIS or other comparable software. Through these menu structures in JavaScript and together with the supplied JavaScript menu code EleProductSite provides the navigation that can be used by the visitors of your website to navigate through the HTML pages that were also generated by EleProductSite.

JavaScript menucode

JavaScript code for the menus is part of EleProductSite. This JavaScript code constitutes and runs the menus inside the browser that visualises your HTML pages generated by EleProductSite. The data for these menus comes from EleProductSite. This data makes up the tree structure of your menus in a form suitable for JavaScript.

Website tree bi-directionally linked to database content

In the case of EleProductSite and EleAIS this website tree of HTML pages is logically and bi-directionally connected to a database so that single pages can be called from database content and specific database content can be called by clicking on hyperlinks on pages. These hyperlinks are typically article numbers.

The structure of the website grouping system

The website grouping system corresponds to the structure of your directory system

EleProductSite was developed to generate and administrate product websites with a rather larger or even a very large number of detailed product HTML pages. These pages are ideally organised and stored in a directory structure (remember that the Windows operating system substantially slows down if there are more than a couple of hundred files inside one directory).

EleProductSite automatically creates a directory system for your HTML pages from and corresponding to the structure of the website grouping system. In fact, it does so by default.

This chapter henceforth explains the correlation between the website grouping system and the structure of your directory system for storing the HTML pages.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Tree structure overview

Like every tree the website publication system (tree) in EleProductSite consists of:

- One root
- Optionally there are several sub-branches each for one content language
- Several branches on different branch levels
- Leafs, which constitute the lowest elements of the tree and which in this case represent the pages.

EleAIS presents this tree hierarchy to the user of an Internet browser viewing your product data.

EleAIS fills one of several different visual components, which allow the user of an Internet browser to navigate inside this tree and to select one or several of the leafs (pages). Such visual components are a tree view or a form with split panes each showing the elements of one level of the tree. For further details please refer to the EleAIS user manual.

Root structure and branches

The structure of the website grouping system fully corresponds to the directory structure on your webserver with the exception of the root of this tree (see next chapter), which is always different.

The entire directory structure is maintained in the website grouping system and it is exported to a corresponding directory structure by the EleProductSite generator. Of course, the EleProductSite generator also creates the directories.

The root of your website grouping system and tree

The EleProductSite grouping system has a root like every tree. Technically, this root consists of some physical disk address, which tells the EleProductSite software as well as the EleAIS software where the HTML pages are physically located. This root is also the left most part of the directory path, which is constructed from the branches of the website grouping system.

The structure of the entire website grouping system and tree corresponds exactly to the structure of the directories holding your HTML pages on your webserver. The only exception to this is a different root, because the directory root (disk drive and top most directory) are typically different on the computer running EleProductSite and on the webserver hardware running EleAIS.

The branches of the website grouping system and tree

The website grouping system provides branches below the root, which enable the user to navigate inside the tree.

The root, the branches and the leafs (pages) altogether constitute the navigation tree, which is presented to the user by EleAIS (or which can also be used by your own webserver software).

Branch levels

The branches are ordered into one or several branch levels. The branch levels create and correspond to the hierarchy of the directory structure where the HTML pages are stored.

Number of branches on one tree level

On every level there can be between one and a couple of hundred different branches. In EleProductSite the number of branches could be practically infinite but in reality the number of branches per level is limited by the limitations of the operating system, which substantially decreases in performance if there are too many directories or files within one directory.

Creating branches and branch levels in the website grouping system

The branches of the website grouping system can be created in one of the following ways:

Creating branches manually

- The branches can be manually created in the website grouping system browser. This is the preferred form when using EleProductSite in conjunction with HTML pages generated by Elecat-P2H and/or as a standalone product when the

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

website product grouping system in EleProductSite is to administrate HTML pages, which were generated by EleExcel or ElePricelist.

Import branches from an ASCII file

- The branch data can be imported from an ASCII file, which allows you to prepare the needed information in some spreadsheet software or to use an existing grouping system from your ERP application. Of course, this data must be enhanced by some fields, which are most likely not supported by your ERP software.

You can create the elements of a grouping tree in an MSeExcel sheet and import this data into Elestore.
This can be much faster than creating the tree levels manually in the Elestore software.

Branches created by Elepub

- If you are using EleProductSite in conjunction with Elepub you can also export an entire Elepub publication structure into the website grouping system and this automatically creates the needed branches. For details see the special chapter on this topic.

Data fields of the website grouping system

This chapter explains the purpose of the data fields of the website grouping system. The website grouping system is one of the most important components of EleProductSite.

The understanding of the data fields of the website grouping system is not only important for entering the data in the user interface but especially if you want to use the website grouping system and its output in conjunction with your own software outside of EleAIS, the webserver software of EleFamily, which also uses these data structures.

EleProductSite path structure for output

EleProductSite expects a path structure for its output. This chapter explains how the path structure is constructed.

This path structure is composed from different components, starting from the root, then adding a language related branch name and the multi-level content branch names.

Except for the root part this path structure is valid both for your local PC as well as for the webserver, which is supplied with data and HTML pages by EleProductSite. The root path will almost always differ, i.e. the data generated by EleProductSite will be stored on a different drive and a different top directory than on your local PC.

Drive and root directory

This is the topmost path corresponding to the root of your website grouping system.

Language path name

See the chapter about Language path name.

Content path names

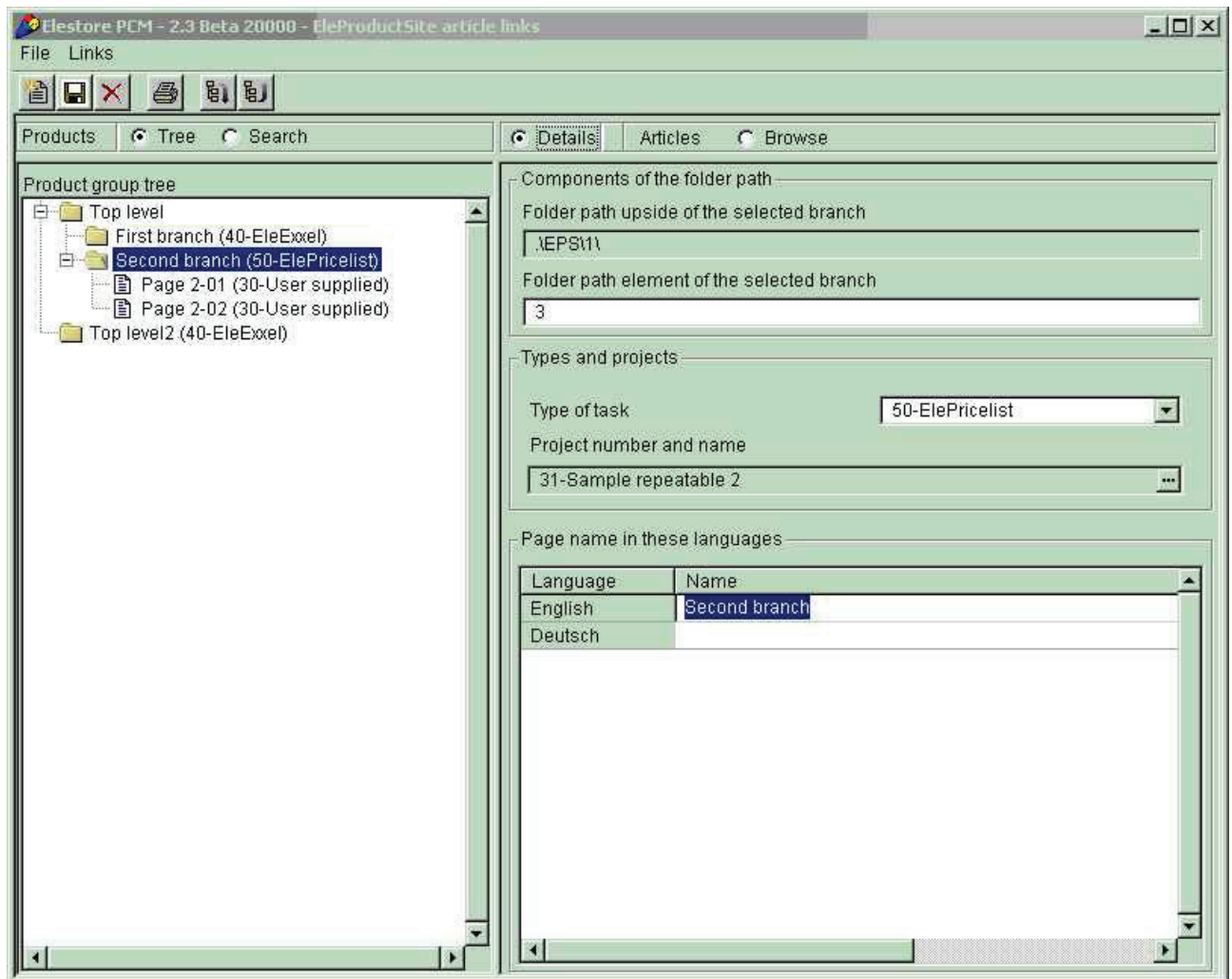
Right inside the language path the content paths are placed. There are as many sub-directory levels as you have defined levels in your website grouping system.

The first path will correspond to the first level branches of your website grouping system. The second level path inside the language directory will correspond to the second level branches of your website grouping system and so forth.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

The website grouping system form

This is the form where the website grouping system is edited:



The three areas of the website grouping system form

This form has three different areas:

- In the left there is the view on the website grouping tree or alternatively a view on searching particular elements in the tree.

The right part of this form contains two very different views:

- In the details view it presents the details known about the one selected grouping branch from the left part of the form.
- The articles browse view shows the articles, which are linked to the currently selected branch of the web grouping tree, and it allows you also to add additional links or remove existing links. This right half is identical with the similar forms in other grouping systems of Elestore.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Menu in the tree view of the website grouping system



Create top branch

This menu item creates a top level branch directly below the root of the grouping tree. Note that the root itself is not visible in this tree view and therefore all top branches below the root are shown on the leftmost position of the tree view.

Create sub branch

This menu item creates a new sub branch below the currently selected branch of the grouping tree. Of course, in order to create a sub-branch you must have another branch selected and this will not work if you select a page, which is a leave of the grouping tree.

Create page

This menu item creates a new page, which is placed below the currently selected branch of the grouping tree. If you add a page to a branch of the grouping tree, this will also mean that no further sub-branches can be added to this branch of the grouping tree, because a branch can either have sub-branches or pages but it cannot have both.

Delete

This menu item deletes the currently selected branch of the grouping tree, of course including all sub-branches and pages, which might be linked to this selected branch.

Delete all

This menu item deletes the entire grouping tree including all sub-branches and pages. You will be asked if you also want to have the content of the EleProductSite root directory deleted that means the generated HTML files.

Delete only HTML files and folders

This menu item will delete only the HTML files and folders in the EleProductSite root directory and it will leave the data of the EleProductSite grouping tree untouched.

Export of tree

This menu item exports the content of the currently selected part of the EleProductSite grouping tree to an ASCII file (text file). This text file contains the data, which makes up the grouping tree such as the ID of the parent level and of the child levels and the language related level and page names.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

This file can be used to edit or enhance the grouping tree or to create a new grouping tree or new elements in some spreadsheet software like MSEXcel. The result of this can then be imported into Elestore.

You can create the elements of a grouping tree in an MSEXcel sheet and import this data into Elestore. Often, this will be much quicker than creating the tree levels manually in the Elestore software.

Open project generator

This menu item opens the EleProductSite project browser where you can select an existing project or create a new one and where you can start the generation process for the currently selected part of the EleProductSite grouping tree.

Please note that the generation of HTML files can either be started by this menu item from the EleProductSite grouping tree or from within the EleProductSite project browser. In the latter case this grouping tree view will first be opened and you will have to select the part of the grouping tree, on which the generator will have to work. Both ways of starting the generation process will have the same results.

Export navigation file

This menu item will create and export JavaScript code and store it into a text file with the extension “js” (for JavaScript). This JavaScript file will contain the external page names and links to the related HTML files for all pages of the currently selected part of the EleProductSite grouping tree.

This JavaScript code can be used for navigation menus in cases where you want to use the generated HTML pages on your website without using EleAIS or any other comparable presentation or WebServer software. This JavaScript code can be easily integrated into JavaScript menus and this will provide the navigation to the users of your website.

There are menus in JavaScript available delivering the entire navigation functionality for the pages generated by EleProductSite. The data for these menus is exported by ProductSite as JavaScript code.

If you need any assistance to implement this please address yourself to our support who will be happy to answer all detailed questions. For any HTML and JavaScript expert it will be obvious and a snap to use this code for providing you with a simple yet very effective navigation for your product website pages that generated by EleProductSite.

Explore directory for selected branch

This menu item opens the Windows Explorer on the directory related to the currently selected branch.

Open page in HTML browser

This menu item opens the currently selected page in your preferred HTML browser. Of course, this works only if you have selected a page and it will not work on the branch, because the branch itself does not consist of an HTML page.

Set root folder setting

This menu item opens the view of the system settings browser on EleProductSite where you can make your entries regarding the paths for the output of EleProductSite. Please refer to the chapter on the system settings for further details. The paths, which you set there, decide about where EleProductSite stores the generated HTML files both on your local PC, on your WebServer and, if applicable, also on the CATpool upload server.

Data fields related to the root of the website grouping system

Overview

This chapter covers only those fields, in which the root of the website grouping system differs from its branches, which implies that some of the fields of their branches also apply to the root.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Drive and home path on your local server

This data field stores the drive and the home path of your entire website grouping system directory structure. This field is mandatory and you cannot run the EleProductSite generator without having entered the physical address of where you want your HTML pages to be stored.

This data field may contain only the topmost (left most) part of the entire directory structure. The content of this data field will be taken by EleProductSite to calculate the full path addresses for all branch levels where the partial paths of the branches are added to the root path. Therefore, when constructing the entire path for a specific directory this drive and home path delivers the first (leftmost) part of the entire directory address.

Note that all of these path addresses must be entered without a leading or an ending path separator (Backslash in the case of Windows).

Drive and home path on your webserver

The drive address and the directory structure on your webserver will in almost every case be different from your local server or local PC. Apart from this distinction the purpose of this data field is identical to the drive and home path on your local server and everything said there applies here as well.

Drive and home path on the CATpool webserver

The drive address and the directory structure on the CATpool server will definitely be different from your webserver.

If you are a CATpool data vendor you receive the drive and home path for your HTML pages on the CATpool webserver from the CATpool support. Please enter this data in this data field. Only if the correct path address is entered here your software will be able to upload its data to the CATpool webserver correctly.

Apart from this distinction the purpose of this data field is identical to the drive and home path on your local server and everything said there applies here as well.

Maximum number of levels

This value is system supplied and it is not an entry field but it is still an important data setting, which is provided hard coded by the EleProductSite software and therefore explained in this chapter.

The website grouping tree can consist of up to 12 levels, which are composed as follows:

- One root level
- Up to 10 nested branch levels
- Plus the lowest level (leafs) for the pages.

Of course, not all branches are forced to have the same number of levels. It just depends on where you add how many branches below the one central root.

We recommend you to use anywhere from two to four levels below the root. In most cases this will be sufficient.

Language path name

EleProductSite is multi-lingual, of course, just like the world is multi-lingual and therefore EleProductSite is capable of creating content for multilingual product websites.

Right below the drive and home directory there is a distinction between the supported content languages. For every content language a main sub directory is created and all of the pages that are generated from content in this language will be stored within this language sub-directory path. There can be one subdirectory for every supported content language.

This language part is also used if you publish only in one single language.

Note that this feature is not yet implemented in this first release of EleProductSite. For the moment you will have to handle the separation of languages manually. This will be added in the next release

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Data fields related to the branches of the website grouping system

Overview

This chapter explains the purpose of the data fields, which are stored for every grouping level branch. Please read also the general overview on this chapter.

Relative path name

This data field stores the part of the path name related to this grouping branch. The content of this data field is used by EleProductSite to automatically generate the entire path for the directory structure. This partial path name will be added to the entire path name at the relative position corresponding to the level number of this branch.

For example, if this value is entered for a first level branch this value will be added immediately to the drive and home path and it will therefore be the second element of the entire path name. This value is entered for a second level branch its value will be added as the third element after the drive and home path and the value for the first level when constructing an entire path name.

A default value for this data field is automatically generated by EleProductSite from the ident number of this grouping branch. Of course, the automatically generated default path name has no significant meaning. You should therefore enter or import a meaningful value if you want your path names to be constructed from meaningful names.

A meaningful name could for example be the name of a chapter corresponding to your paper product catalogue or it could also be the visible external name of his branch in your primary language.

The characters entered into this data field must be legal characters for naming directories in the supported operating system. Please remember that not all characters are allowed in path names. You better avoid blanks inside path names as they often cause problems.

Type of link

This selection field is probably the most important setting to make for a branch in the EleProductSite grouping system. This setting defines the actions that the EleProductSite generator performs on this branch.

The following settings are available

0-no task

This is the default setting and it means that no task has yet been assigned to this branch. Therefore, such branches are skipped by the EleProductSite generator and no data will be generated or exported for such branches.

10-Elepub

This setting indicates that the data for this branch is supplied by Elepub. EleProductSite will skip such branches. You have to start Elepub to generate and export pages and data for this branch from within Elepub.

There is intentionally no enforced parallelism between the structure of a publication in Elepub and the structure of the website grouping system, because this leaves you much more freedom and possibilities to mix pages from Elepub with pages from different other sources. In other words, it is up to your decision to place the pages of a whole Elepub publication, or a chapter, or a section below one of the branches of the EleProductSite grouping system. This is one of the reasons why you have to start the export of the pages for such branches from within Elepub.

This is also the reason why you have to select a branch in the EleProductSite grouping system when starting an export of pages to EleProductSite within Elepub. Please refer to the respective chapter in Elepub where this is explained in detail.

We recommend you to only equip high-level branches with this setting and to start the generation of Elepub of pages for such a branch from within Elepub.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

[20-Elecat-P2H](#)

This setting defines that the pages for this branch are generated by Elecat-P2H by converting them from PDF to HTML. They are then assigned by Elecat-P2H to this branch of the EleProductSite grouping system. This must be started in Elecat-P2H. Therefore, EleProductSite will skip a branch with this setting completely and nothing will be done.

[30-user supplied](#)

This setting defines that all of the pages for this branch are supplied by you from some external source and not from any member of EleFamily. As a consequence, EleProductSite will skip a branch with this setting completely and nothing will be done.

[40-EleExxel](#)

This setting defines that the pages for this branch are to be generated by EleExxel. As a consequence, the settings below are activated and you have to select a task from the EleExxel project list, which is to be executed by EleProductSite to generate the pages and the data for this branch.

[50-ElePricelist](#)

The same applies as to the setting for EleExxel only with the difference that a project from ElePricelist is selected and executed for this branch.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Data fields related to the pages of the website grouping system

Overview

This chapter explains the purpose of some data fields, which are stored for every page in the website grouping system. The pages constitute the leafs of the website grouping system. Please read also the general overview on this chapter.

External page name

The external page name is the page name that the user sees when navigating the website tree in EleAIS or in some other software using the data generated by EleProductSite. Of course, the external page name is language dependent and one such page name exists for every content language.

The value of the external page name is either manually entered when editing the website grouping system or it has been imported from one of the different sources supported by EleProductSite such as:

- pages converted from PDF to HTML by Elecat-P2H,
- HTML pages generated by EleExxel or ElePricelist,
- or HTML pages generated by Elepub.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

EleProductSite projects and their settings

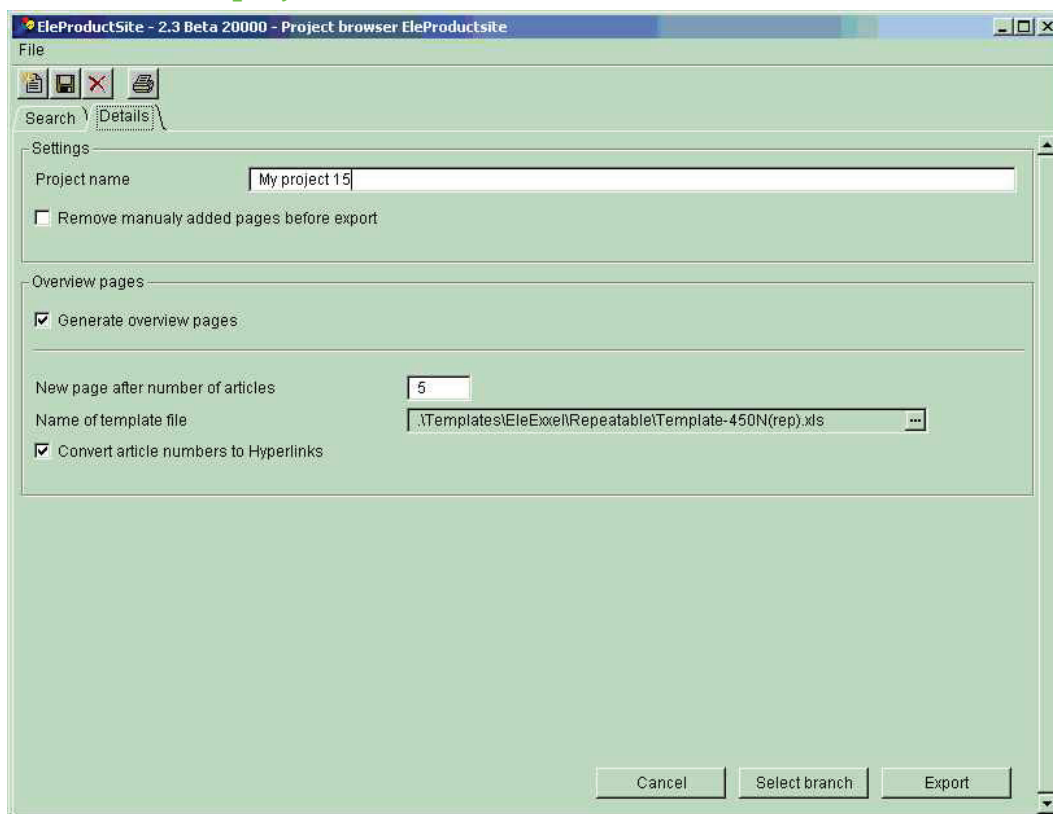
Overview and purpose of projects in EleProductSite

Like in many other Elestore modules and EleFamily products the projects in EleProductSite store all the parameters needed to execute a particular EleProductSite task. This allows an easy reuse of settings.

EleProductSite project table

From the EleProductSite project table you can select one of the stored projects for a simple reuse.

EleProductSite project details form



The screenshot shows a software window titled "EleProductSite - 2.3 Beta 20000 - Project browser EleProductsite". It has a menu bar with "File" and a toolbar with icons for file operations. Below the toolbar are two tabs: "Search" and "Details", with "Details" being the active tab. The "Details" tab is divided into two sections: "Settings" and "Overview pages".

Settings section:

- "Project name": A text input field containing "My project 15".
- "Remove manually added pages before export": An unchecked checkbox.

Overview pages section:

- "Generate overview pages": A checked checkbox.
- "New page after number of articles": A text input field containing "5".
- "Name of template file": A text input field containing ".\\Templates\\EleExel\\Repeatable\\Template-450N(rep).xds" with a browse button (three dots) to its right.
- "Convert article numbers to Hyperlinks": A checked checkbox.

At the bottom of the window are three buttons: "Cancel", "Select branch", and "Export".

Remarks on the EleProductSite project details form

The project details form of EleProductSite looks rather sparse. Assuming from this that EleProductSite could also be called a sparse product would be completely wrong.

The power and the flexibility of EleProductSite come from the fact that EleProductSite manages and arranges the systematic use of several other entities from within EleFamily. Using an analogy from the business world one could call EleProductSite a holding company, which administrates and sits on top of several autonomous active companies, which are represented by products like EleExxel, EleProductSite, or Elepub.

EleProductSite manages and drives these other software products and orders their output. Much of how this is done is determined by the website grouping system and its entries.

Therefore, in order to understand the functionality and the enormous flexibility of EleProductSite one has to know and understand the website grouping system as well as the underlying software products like EleExxel, EleProductSite, and Elepub etc

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

The settings of an EleProductSite project

Project name

The field project name stores a text describing the purpose of this project. You should select some meaningful explanation of what this project does so that it will be easy for you or other users to select and re-use this project without having to look at all the settings.

Remove manually added pages before export

If you activate this check box EleProductSite will automatically remove all manually added pages before starting its export.

Generate overview pages

If you activate this setting EleProductSite will automatically generate additional overview pages, which contain the data objects (articles) with or without links to the generated HTML details pages that contain these article objects.

These overview pages can be presented to the user of an Internet browser as a way of navigating through the data content of your product website and for selecting particular article details pages in HTML.

New page after number of articles

This checkbox lets you decide how many article objects you want to be output on one single HTML details page.

Essentially, you can select between just one article or a specific other number of article objects so that one page would contain the details of several article objects. This is more a general and conceptual question whether or not you want article details pages with just one article object or if you want bigger pages containing the data and content of multiple article objects.

Please note that this also depends on the type of projects that you have selected for the processed web grouping system branch. In other words, if the selected articles are linked to single and non-repeatable page templates it would not make sense to have more than just one article being generated on one page. Therefore, the number of article objects that is entered here must correspond to the underlying project.

Name of template file

You can select here some template file, which will be used for generating the overview pages. This can be either an MSExcel file, which will be processed by EleExxel, or an MSWord file, which will be processed by ElePricelist.

The layout and the attributes, which are supposed to be output for each article object on the overview pages, is defined by the structure of this template file. For details about this template file please refer to the chapters on EleExxel or ElePricelist, because it is technically EleExxel or ElePricelist, that are used to generate these overview pages.

Convert article numbers to hyperlinks

If you activate this checkbox the article numbers on the generated overview pages will be converted to hyperlinks, which point to the details pages. These hyperlinks enable the user of an Internet browser to select an article from the overview page and to have the corresponding article details page opened.

If you deactivate this checkbox you can generate overview pages without hyperlinks, which therefore will not lead the user to some underlying details pages. This should be the exception but it can nevertheless be occasionally wanted.

The EleProductSite generator

Overview

The EleProductSite generator performs various batch operations based on the EleProductSite website grouping system and the settings that you have made to its branches. Many of the operations performed by the EleProductSite generator actually refer to data and/or procedures, which are part of other EleFamily products or modules like EleExxel, ElePricelist or Elepub whose projects are embedded into the parameters of the branches of the website grouping system.

The EleProductSite generator also starts the execution of projects in EleExxel and ElePricelist and processes and stores their results. This is one of the reasons why EleExxel and ElePricelist are fully embedded by EleProductSite.

Starting the EleProductSite generator

The EleProductSite generator can be started via the right mouse menu on every level of the website grouping system and also on the root.

The EleProductSite generator will loop through all selected grouping levels and within every selected grouping level it will also loop through all of its sub-grouping levels. For each of these levels the EleProductSite generator will execute the task, which is connected with each particular grouping level.

Executing procedures linked to a website grouping level

Depending on the type of grouping level the execution of specific tasks can be activated for every grouping level. This grouping level task is executed on the EleProductSite generator is started on this particular grouping level or on one of its super levels.

The type of task, which is executed, depends on the level type, which was assigned by the user to each grouping level.

Available website grouping level types

Every grouping level of the website grouping system can be assigned to a specific task. The available tasks are described henceforth:

Website grouping level of type unassigned

This is the default type of grouping level and unassigned means that there is no task assigned to this grouping level. All grouping levels, which are the holders of sub grouping levels, are automatically unassigned. As a consequence, tasks can only be assigned to the lowest levels of the grouping system.

This type of grouping level is also the preferred one if the user wishes to add HTML pages to this website, which were produced by some third party software or HTML editor outside EleFamily. Therefore, if you want to add your already existing HTML pages to your website operated by EleAIS you simply add these pages manually and physically into the directory, which is related to one of several such grouping levels of this type.

Website grouping level of type EleExxel

A grouping level of this type is linked with a project in EleExxel and this project is automatically executed when the EleProductSite generator is started on grouping levels of this type. The HTML pages generated by EleExxel are automatically exported into the directory, which is related to this grouping level. All the other settings are also taken from the EleExxel project. This is why it is possible to automatically generate several or many EleExxel projects by simply selecting one or several grouping levels or even the route of the website grouping system of EleProductSite.

Website grouping level of type ElePricelist

This is identical to a grouping level of type EleExxel only that the selected project refers to an ElePricelist project, which can also embed one or several EleExxel projects into its output.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Website grouping level of type Elepub

A grouping level of this type is linked to a chapter or a section of Elepub. When the EleProductSite generator is started on such a grouping level EleProductSite will automatically start the HTML export module of Elepub, which then exports the Elepub pages contained in this chapter or section into the directory related to this grouping level.

Please note that it is up to the user to decide whether to link a chapter or a section to a grouping level. This flexibility enables you to modify the tree structure of the HTML pages compared to the structure of Elepub so that websites generated by EleProductSite from data in Elepub can be structured differently compared to the original Elepub publications. This often makes sense, because a paper publication is typically divided into a smaller number of chapters while the equivalent website would be divided into a tree of more levels and therefore into a larger number of branches and sub-directories.

Applying this grouping level requires that Elepub is part of your installation.

Website grouping level of type Elecat-P2H

A grouping level of this type receives its data from the Elecat PDF to HTML generator. This also means that a grouping level of this type will be skipped during the automatic execution of the EleProductSite generator, because the export of HTML must be started from within the Elecat-P2H software with various settings individually prepared for every export procedure.

This is why in the case of this type of grouping level the activity must be started from within the Elecat-P2H software and therefore grouping level entries of this type are skipped by the EleProductSite generator.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Exporting an Elepub publication to EleProductSite

Overview

Elepub is the database publishing system of EleFamily and it is together with Elestore the most sophisticated and important component of the EleFamily software products. Elepub was primarily designed to generate paper product catalogs but it is equally well suitable for generating pages for websites, for electronic product catalogs and for other electronic media.

Elepub generates HTML pages directly by itself without any need for additional software. However, in cases where the HTML pages generated by Elepub are to be mixed with HTML pages from other sources and/or where the user wants a complex website page structure to be created it will be helpful to use Elepub's HTML generation capabilities in conjunction with EleProductSite.

This co-operation between EleProductSite and Elepub is explained in this chapter.

Structural differences between an Elepub publication and a product website

Structures of paper product catalogs

First of all, one has to understand the natural structural differences between a typical Elepub publication and a typical product website.

Elepub was primarily designed to generate paper product catalogs. The layout structures in paper product catalogs are typically multiple times more complex than the layouts used for product websites. Therefore, generating product pages in HTML is the comparatively simpler task as far as the complexity of layout structures is concerned.

However, the tree structure of a product website cannot be found in a paper catalog due to the natural limitations of paper. A typical tree structure of a product website is different in several aspects from the structure of a paper publication. Therefore and in order to make the use of Elepub much easier, Elepub was intentionally designed to support the specific organizational structure of paper publications.

Elements of an Elepub publication

An Elepub publication consists of the following components:

- The Elepub object database stores any number of publications.
- One Elepub publication consists of one or any number of chapters.
- One chapter consists of one or any number of sections, which are organizational entities whose purpose is in subdividing a chapter into one or several autonomous parts. The primary purpose of a section is to define a specific number of data objects (articles, images etc) and the sequence, in which they are published. Sections are internal organizational elements that are in most cases not visible to the reader of the final product catalog.
- Every section consists of one or any number of pages.

Websites have different structures

This structure is not only ideally suitable for paper product catalogs but it can also just as well be used to generate the pages for a product website or any other electronic media. However, in the case of electronic media the user visible organization of the pages might be somewhat different from the typical publication structure supported by Elepub.

One of several ways of achieving this is by using the website grouping system supported by EleProductSite. How this is achieved is the subject of this chapter.

Structure of a product website with respect to Elepub

Product websites are always constructed as a tree. Often, the data is ordered in a multi-dimensional tree in so far as there are several different ways offered to the user to wade through the pages of the product website. Very often, several products and/or pages are published in different branches of such a multidimensional tree. This is intentional in order to present products wherever it makes sense.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

A simple example: a spare part may be published as an additional product to all the machines to which it serves as a spare part. Additionally, this spare part would be published as an autonomous product in the spare part chapter. As a result of this, this spare part product would be published multiple times in different parts of the website tree.

In a paper product catalogue essentially the same problem exists but there this is typically solved by applying layouts of very different complexity to this spare part. In other words, whether the spare part is published as part of a machine, to which it is a component, this spare part product would only be mentioned with its minimal information and maybe with the link to its main page. All the details of this product are only published once in the paper catalogue, because everything else would make the paper catalogue much thicker, much more expensive and less user-friendly.

Just the opposite situation exists in a product website, which is ideally suitable to present all the technical details of this spare part wherever the spare part is referred to. Space costs no money on a website, nor does the traffic (or at least the costs are so minimal that it is not worth discussing them).

These requirements of a product website can be solved by links, of course, but they can also be solved by redundant data, which is not necessarily negative in this case. This redundancy can be achieved by applying products to several main branches of the website tree. Similar requirements exist wherever products are grouped to product family's and where components are presented as autonomous products and additionally as part of several main or family products. All of these frequent situations require redundant presentation of product details and layout and often the different situations also require different layouts.

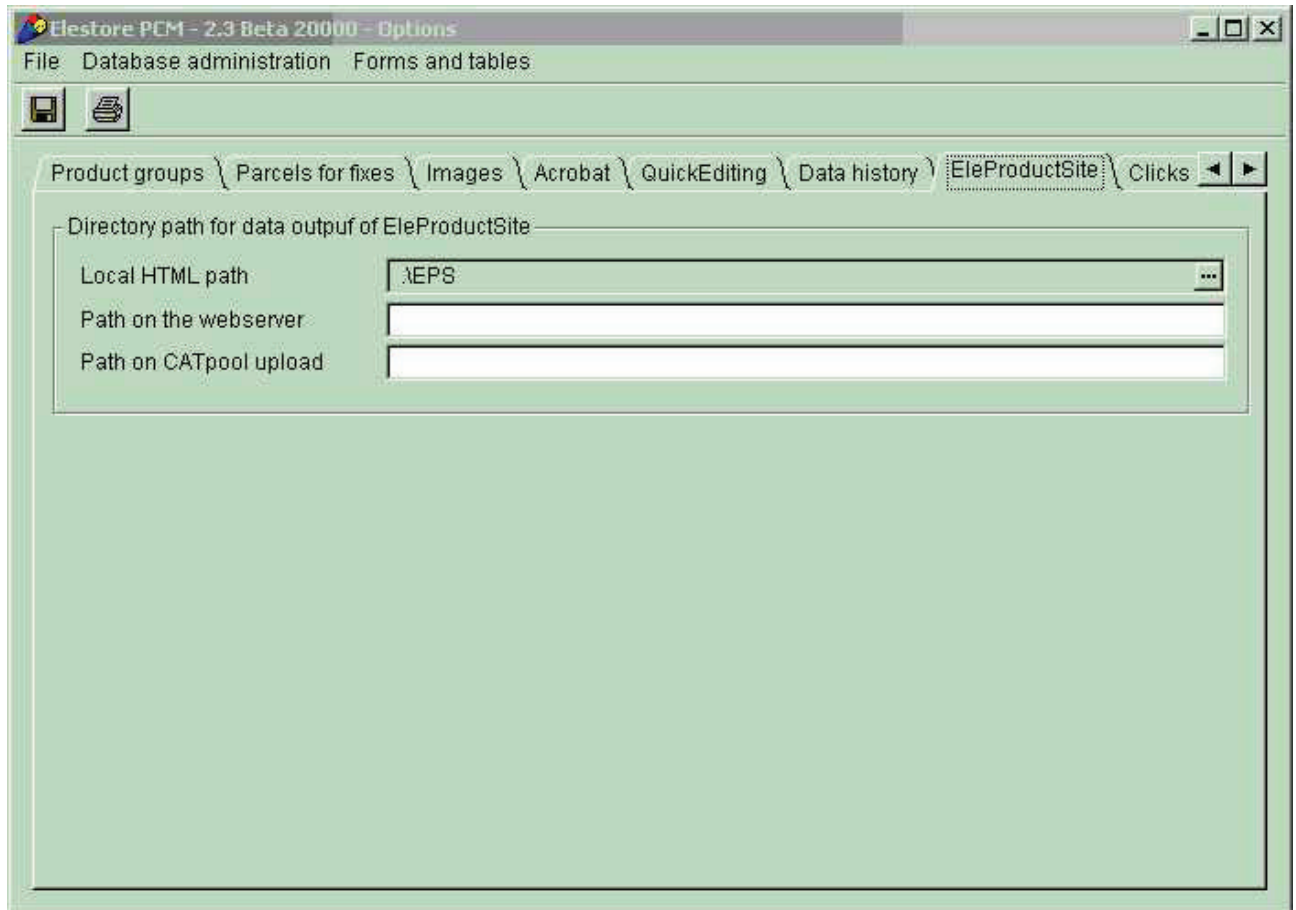
The EleProductSite website grouping system is the answer and the solution to perfectly meet these and many other requirements and to easily group the output of Elepub according to the various requirements to structure a product website.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

System settings for EleProductSite

The EleProductSite related form in system settings

The fields of this tab in the system settings browser are related to EleProductSite:



Purpose of the EleProductSite related system settings

When the EleProductSite generator produces HTML it must know the folder address where these HTML files are to be stored. This folder address is the root, below which EleProductSite will create its directory structure for the HTML pages.

The same applies when EleProductSite is supposed to upload these files to your webserver and/or to the CATpool webserver. In all of these cases the needed folder root paths are set in this system settings form. For the usage of these values please refer to the other chapters of the EleProductSite user manual where this is explained in detail.

Local HTML path

In this field you select the drive and the root folder for the directory structure, where the EleProductSite generator is supposed to store the HTML files that it generates. Please press the little button in the right corner of the entry field and select the topmost folder (directory) and store it in this field, below which EleProductSite can create its own folder structure. For further details please refer to the chapter about the web grouping system in the EleProductSite manual.

Path on the webserver

Please enter into this data field the root path on your webserver for the directory structure, below which EleProductSite stores the HTML files during the upload.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

Unfortunately, there is no possibility for EleProductSite to check if you have entered the correct address, because the access to your webserver is done via FTP and it is therefore not identical to accessing a local disk drive inside your LAN. This is why you have to enter the address instead of selecting it as you can do for the local root directory. Therefore, please take care that the address is entered correctly.

If you don't know this address or if you are having problems with entering it then please consult our support people who will help you.

Path on CATpool upload

This is only for CATpool member. This field is identical to the path on your webserver with the difference that this is the path on the CATpool webserver for the upload of HTML pages. If you are a CATpool member you have received an e-mail from the CATpool support telling you the name of your upload path. Please enter this correctly into this field.

All the rest of what is written about the path on your own webserver is also true for CATpool.

Error messages in EleProductSite

Overview

This is a list of some of the most important error messages in EleProductSite with an explanation of their causes. The error messages are identified by their text key, which is shown below the actual user notification that comes up.

n_infoExportNoTemplateSelected

You have not yet selected a template file and this is why the application cannot start.

You have probably selected a branch of the grouping tree, which is the lowest of all levels. This can for example be indicated by the fact that this level already has pages linked to it.

Please note that you also cannot add a sub-branch if the related branch has already pages below itself. A branch can either have one or several sub branches or pages but not both types.

n_infoEpsGrpLevCanNotAddBranchToLowestBranchWithPages

There is something wrong with the path that you have entered or selected, because the application software cannot construct the directory path using your entry. Please correct your entry and enter or selected a valid path.

n_infoEpsGrpLevCanNotChangeConstructedPathFromParentHieararchy

It is not allowed to move a group to another parent.

n_infoEpsGrpLevCanNotMoveGroupsToOtherParentGroup

Moving one group level to another parent group is not allowed in this case. You will have to delete this group level and create it again below the other parent group.

n_infoEpsGrpLevDriverSysAlreadyOpenedUnsavedChangesMayBeLost

Due to this action it can occur that some of your changes, which were not yet written to the database, will be lost.

n_infoEpsGrpLevMustSelectSingleBranch

EleProductSite generator you may only selected one branch and there is currently no or more than one branch selected.

n_infoEpsGrpLevMustSelectSingleLowestBranch

When adding a new page you must have one single branch of the lowest level selected. This is currently not the case.

n_infoEpsGrpLevMustSelectSinglePage

In order to open an HTML page you must select one single HTML page file only. It can be the case that you have either selected more than one page all that you have selected a branch, which does not represent a page.

n_infoEpsGrpLevMustSelectSomeBranch

You have not selected any branch and you can therefore not link any articles, because the application must know to which branch you want to link the selected data objects.

Note: This manual is “work in progress” and all contents can be subject to change without any notice. This manual also contains concepts of possible future features, but there is no guarantee when or that these concepts will be implemented.

n_infoEpsGrpLevPageCanNotAddSubBranchLimitReached

You cannot add another sub-branch, because the maximum allowed number of levels would be exceeded. For practical reasons we have hard coded a maximum of 12 levels for the publication tree. Actually, we doubt that any user will ever arrive at this place and if you do then give us a note, because you have won the jackpot! *ggg*

n_infoEpsGrpLevPageCanNotHaveSubLevel

One cannot add a sub-branch to a page. A page constitutes the leaf in the publication tree.

n_infoEpsGrpLevReorderingInProgress

The reordering of the publication tree is currently performed. Please wait a minute.